

PP/018/Sms ke-.../Revisi ke-



LABORATORIUM
TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS AHMAD DAHLAN



PETUNJUK PRAKTIKUM

Pengenalan POLA

Penyusun

Dr. Abdul Fadhil, M.T

Ahmad Azhari, S.Kom., M.Eng

Miftahurrahma Rosyda, S.Kom., M.Eng

2020

KATA PENGANTAR

Puji syukur kami panjatkan kehadiran Allah SWT yang telah melimpahkan rahmat dan karunia-Nya, sehingga dapat diselesaikannya Petunjuk Praktikum ini. Sholawat dan salam semoga terlimpah kepada baginda Nabi Muhammad saw beserta keluarga, shohabat dan para pengikutnya.

Petunjuk Praktikum Pengenalan Pola ini disusun dengan harapan dapat memberikan kemudahan bagi mahasiswa untuk memahami dasar-dasar dalam membangun sebuah sistem pengenalan pola. Berbagai contoh aplikasi diberikan yang meliputi metode ekstraksi ciri maupun teknik-teknik klasifikasi. Ucapkan terima kasih yang sebesar-besarnya diberikan kepada pihak semua pihak yang telah membantu dalam proses penyusunan petunjuk praktikum ini.

Akhirnya, tiada gading yang tak retak, kritik dan saran membangun kami harapkan untuk lebih sempurnanya petunjuk praktikum ini. Semoga ilmu yang diperoleh dari membaca dan mempraktekkan petunjuk praktikum ini dapat bermanfaat dan barokah.

Yogyakarta, Januari 2020

Penyusun

DAFTAR PENYUSUN

Ahmad Azhari, S.Kom., M.Eng.

Miftahurrahma Rosyda, S.Kom., M.Eng.

Dr. Abdul Fadlil, M.T.

HALAMAN REVISI

Yang bertanda tangan di bawah ini:

Nama : Ahmad Azhari, S.Kom., M.Eng.

NIK/NIY : 60160863

Jabatan : Dosen Pengampu

Dengan ini menyatakan pelaksanaan Revisi Petunjuk Praktikum Pengenalan Pola untuk Program Studi Teknik Informatika telah dilaksanakan dengan penjelasan sebagai berikut:

No	Keterangan Detail Revisi (Per Pertemuan)	Tanggal Revisi	Nomor Modul
1	a. Menambahkan teori tentang pengenalan pola pada bagian langkah praktikum b. membenarkan kode program		

Yogyakarta, 22 Januari 2020

Penyusun

Ahmad Azhari, S.Kom., M.Eng

NIK/NIY.60160863

HALAMAN PERNYATAAN

Yang bertanda tangan di bawah ini:

Nama : Ahmad Azhari, S.Kom., M.Eng.

NIK/NIY : 60160863

Jabatan : Dosen Pengampu

Menerangkan dengan sesungguhnya bahwa Petunjuk Praktikum ini telah direview dan akan digunakan untuk pelaksanaan praktikum di Semester Genap Tahun Akademik 2019/2020 di Laboratorium Komputasi Dasar, Program Studi Teknik Informatika, Fakultas Teknologi Industri, Universitas Ahmad Dahlan.

Yogyakarta, 22 Januari 2020

Mengetahui,

Ketua Kelompok Keilmuan Sistem Cerdas

Kepala Laboratorium Komputasi Dasar

Dewi Soyusiawaty, S.T., M.T.

NIK/NIY. 60040497

Adhi Prahara, S.Si., M.Cs.

NIK/NIY. 60150841

VISI DAN MISI PRODI TEKNIK INFORMATIKA

VISI

Menjadi Program Studi Informatika yang diakui secara internasional dan unggul dalam bidang Informatika serta berbasis nilai-nilai Islam.

MISI

1. Menjalankan pendidikan sesuai dengan kompetensi bidang Informatika yang diakui nasional dan internasional
2. Meningkatkan penelitian dosen dan mahasiswa dalam bidang Informatika yang kreatif, inovatif dan tepat guna.
3. Meningkatkan kuantitas dan kualitas publikasi ilmiah tingkat nasional dan internasional
4. Melaksanakan dan meningkatkan kegiatan pengabdian masyarakat oleh dosen dan mahasiswa dalam bidang Informatika.
5. Menyelenggarakan aktivitas yang mendukung pengembangan program studi dengan melibatkan dosen dan mahasiswa.
6. Menyelenggarakan kerja sama dengan lembaga tingkat nasional dan internasional.
7. Menciptakan kehidupan Islami di lingkungan program studi.

TATA TERTIB LABORATORIUM TEKNIK INFORMATIKA

DOSEN/KOORDINATOR PRAKTIKUM

1. Dosen harus hadir saat praktikum minimal 15 menit di awal kegiatan praktikum dan menandatangani presensi kehadiran praktikum.
2. Dosen membuat modul praktikum, soal seleksi asisten, pre-test, post-test, dan responsi dengan berkoordinasi dengan asisten dan pengampu mata praktikum.
3. Dosen berkoordinasi dengan koordinator asisten praktikum untuk evaluasi praktikum setiap minggu.
4. Dosen menandatangani surat kontrak asisten praktikum dan koordinator asisten praktikum.
5. Dosen yang tidak hadir pada slot praktikum tertentu tanpa pemberitahuan selama 2 minggu berturut-turut mendapat teguran dari Kepala Laboratorium, apabila masih berlanjut 2 minggu berikutnya maka Kepala Laboratorium berhak mengganti koordinator praktikum pada slot tersebut.

PRAKTIKAN

1. Praktikan harus hadir 15 menit sebelum kegiatan praktikum dimulai, dan dispensasi terlambat 15 menit dengan alasan yang jelas (kecuali asisten menentukan lain dan patokan jam adalah jam yang ada di Laboratorium, terlambat lebih dari 15 menit tidak boleh masuk praktikum & dianggap Inhal).
2. Praktikan yang tidak mengikuti praktikum dengan alasan apapun, wajib mengikuti INHAL, maksimal 4 kali praktikum dan jika lebih dari 4 kali maka praktikum dianggap GAGAL.
3. Praktikan harus berpakaian rapi sesuai dengan ketentuan Universitas, sebagai berikut:
 - a. Tidak boleh memakai Kaos Oblong, termasuk bila ditutupi Jaket/Jas Almamater (Laki-laki / Perempuan) dan Topi harus Dilepas.
 - b. Tidak Boleh memakai Baju ketat, Jilbab Minim dan rambut harus tertutup jilbab secara sempurna, tidak boleh kelihatan di jidat maupun di punggung (khusus Perempuan).
 - c. Tidak boleh memakai baju minim, saat duduk pun pinggang harus tertutup rapat (Laki-laki / Perempuan).
 - d. Laki-laki tidak boleh memakai gelang, anting-anting ataupun aksesoris Perempuan.
4. Praktikan tidak boleh makan dan minum selama kegiatan praktikum berlangsung, harus menjaga kebersihan, keamanan dan ketertiban selama mengikuti kegiatan praktikum atau selama berada di dalam laboratorium (tidak boleh membuang sampah sembarangan baik kertas, potongan kertas, bungkus permen baik di lantai karpet maupun di dalam ruang CPU).
5. Praktikan dilarang meninggalkan kegiatan praktikum tanpa seizin Asisten atau Laboran.
6. Praktikan harus meletakkan sepatu dan tas pada rak/loker yang telah disediakan.
7. Selama praktikum dilarang NGENET/NGE-GAME, kecuali mata praktikum yang membutuhkan atau menggunakan fasilitas Internet.
8. Praktikan dilarang melepas kabel jaringan atau kabel power praktikum tanpa sepengetahuan laboran
9. Praktikan harus memiliki FILE Petunjuk praktikum dan digunakan pada saat praktikum dan harus siap sebelum praktikum berlangsung.

10. Praktikan dilarang melakukan kecurangan seperti mencontek atau menyalin pekerjaan praktikan yang lain saat praktikum berlangsung atau post-test yang menjadi tugas praktikum.
11. Praktikan dilarang mengubah setting software/hardware komputer baik menambah atau mengurangi tanpa permintaan asisten atau laboran dan melakukan sesuatu yang dapat merugikan laboratorium atau praktikum lain.
12. Asisten, Koordinator Praktikum, Kepala laboratorium dan Laboran mempunyai hak untuk menegur, memperingatkan bahkan meminta praktikan keluar ruang praktikum apabila dirasa anda mengganggu praktikan lain atau tidak melaksanakan kegiatan praktikum sebagaimana mestinya dan atau tidak mematuhi aturan lab yang berlaku.
13. Pelanggaran terhadap salah satu atau lebih dari aturan diatas maka Nilai praktikum pada pertemuan tersebut dianggap 0 (NOL) dengan status INHAL.

ASISTEN PRAKTIKUM

1. Asisten harus hadir 15 Menit sebelum praktikum dimulai (konfirmasi ke koordinator bila mengalami keterlambatan atau berhalangan hadir).
2. Asisten yang tidak bisa hadir WAJIB mencari pengganti, dan melaporkan kepada Koordinator Asisten.
3. Asisten harus berpakaian rapi sesuai dengan ketentuan Universitas, sebagai berikut:
 - a. Tidak boleh memakai Kaos Oblong, termasuk bila ditutupi Jaket/Jas Almamater (Laki-laki / Perempuan) dan Topi harus Dilepas.
 - b. Tidak Boleh memakai Baju ketat, Jilbab Minim dan rambut harus tertutup jilbab secara sempurna, tidak boleh kelihatan di jidat maupun di punggung (khusus Perempuan).
 - c. Tidak boleh memakai baju minim, saat duduk pun pinggang harus tertutup rapat (Laki-laki / Perempuan).
 - d. Laki-laki tidak boleh memakai gelang, anting-anting ataupun aksesoris Perempuan.
4. Asisten harus menjaga kebersihan, keamanan dan ketertiban selama mengikuti kegiatan praktikum atau selama berada di laboratorium, menegur atau mengingatkan jika ada praktikan yang tidak dapat menjaga kebersihan, ketertiban atau kesopanan.
5. Asisten harus dapat merapikan dan mengamankan presensi praktikum, Kartu Nilai serta tertib dalam memasukan/Input nilai secara Online/Offline.
6. Asisten harus dapat bertindak secara profesional sebagai seorang asisten praktikum dan dapat menjadi teladan bagi praktikan.
7. Asisten harus dapat memberikan penjelasan/pemahaman yang dibutuhkan oleh praktikan berkenaan dengan materi praktikum yang diasistensi sehingga praktikan dapat melaksanakan dan mengerjakan tugas praktikum dengan baik dan jelas.
8. Asisten tidak diperkenankan mengobrol sendiri apalagi sampai membuat gaduh.
9. Asisten dimohon mengkoordinasikan untuk meminta praktikan agar mematikan komputer untuk jadwal terakhir dan sudah dilakukan penilaian terhadap hasil kerja praktikan.
10. Asisten wajib untuk mematikan LCD Projector dan komputer asisten/praktikan apabila tidak digunakan.
11. Asisten tidak diperkenankan menggunakan akses internet selain untuk kegiatan praktikum, seperti Youtube/Game/Medsos/Streaming Film di komputer praktikan.

LAIN-LAIN

1. Pada Saat Responsi Harus menggunakan Baju Kemeja untuk Laki-laki dan Perempuan untuk Praktikan dan Asisten.
2. Ketidakhadiran praktikum dengan alasan apapun dianggap INHAL.
3. Izin praktikum mengikuti aturan izin SIMERU/KULIAH.
4. Yang tidak berkepentingan dengan praktikum dilarang mengganggu praktikan atau membuat keributan/kegaduhan.
5. Penggunaan lab diluar jam praktikum maksimal sampai pukul 21.00 dengan menunjukkan surat izin dari Kepala Laboratorium Prodi Teknik Informatika.

Yogyakarta, 22 Januari 2020

Kepala Laboratorium Komputasi Dasar

Adhi Prahara, S.Si., M.Cs.

NIK/NIY. 60150841

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR PENYUSUN	ii
HALAMAN REVISI	iii
HALAMAN PERNYATAAN	iv
VISI DAN MISI PRODI TEKNIK INFORMATIKA	v
TATA TERTIB LABORATORIUM TEKNIK INFORMATIKA.....	vi
DAFTAR ISI.....	ix
DAFTAR GAMBAR.....	x
PRAKTIKUM 1: PENGANTAR MATLAB	1
PRAKTIKUM 2: PEMROSESAN AWAL.....	18
PRAKTIKUM 3: EKSTRAKSI CIRI.....	25
PRAKTIKUM 4: KLASIFIKASI DENGAN GARIS LINEAR.....	28
PRAKTIKUM 5: KLASIFIKASI DENGAN METRIK JARAK.....	43
PRAKTIKUM 6: SISTEM PENGENALAN WAJAH	48
PRAKTIKUM 7: JARINGAN SYARAF TIRUAN UNTUK PENGENALAN POLA	55
PRAKTIKUM 8: ALGORITMA PERCEPTRON UNTUK PENGENALAN POLA	0
PRAKTIKUM 9: ALGORITMA BACKPROPAGATION UNTUK PENGENALAN POLA	9
PRAKTIKUM 10: ALGORITMA LVQ UNTUK PENGENALAN POLA	17
DAFTAR PUSTAKA	21

DAFTAR GAMBAR

Gambar 1. 1 Tampilan Command Window.....	Error! Bookmark not defined.
Gambar 1. 2 Layar Editor	5
Gambar 1. 3. Hasil output coba.m	7
Gambar 1. 4 Hasil skrip kasus1.m	10
Gambar 1. 5. Hasil skrip kasus2.m	12
Gambar 1. 6 Hasil skrip kasus5.m	14
Gambar 1. 7 Hasil Output kasus6.m	15
Gambar 1. 8 Hasil skrip kasus7.m	16
Gambar 3. 1 Citra tesktur makrosutruktur	26
Gambar 3. 2 citra teksutr mikrostruktur.....	26
Gambar 4. 1 Fungsi keputusan untuk mengklasifikasi dua jenis pola.....	29
Gambar 6. 1. Blok Diagram Sistem Pengenalan Wajah.....	49
Gambar 7. 1. Arsitektur Jaringan Syaraf Tiruan Sederhana	56
Gambar 9. 1 Arsitektur Backpropagation	10

PRAKTIKUM 1: PENGANTAR MATLAB

Pertemuan ke : 1

Total Alokasi Waktu : 90 menit

- Pre-Test : 10 menit
- Praktikum : 65 menit
- Post-Test : 15 menit

Total Skor Penilaian : 100%

- Pre-Test : 25 %
 - Praktikum : 40 %
 - Post-Test : 35%
-

1.1. TUJUAN DAN INDIKATOR CAPAIAN

Setelah mengikuti praktikum ini mahasiswa diharapkan:

1. Mahasiswa mampu menjelaskan operasi dasar menggunakan Matlab

Indikator ketercapaian diukur dengan: (sesuaikan dengan RPS)

1. Membuat program sederhana menggunakan MATLAB dengan *.m file
2. Memahami statemen operasi kalang dan mampu membuat grafik.

1.2. TEORI PENDUKUNG

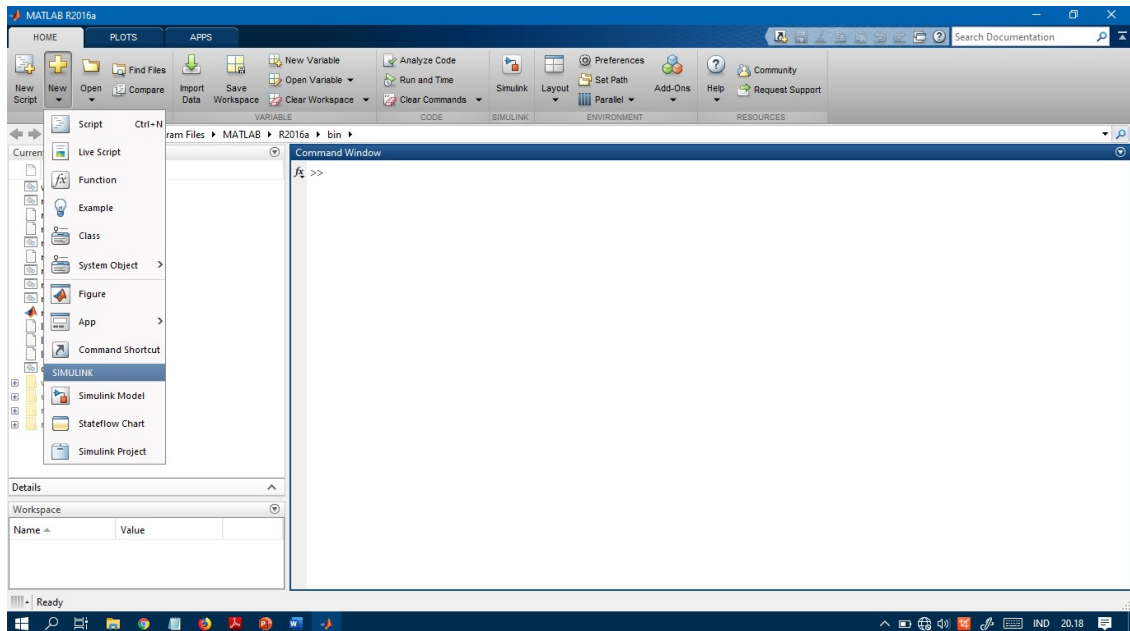
MATLAB (**MAT**rix **LAB**oratory) merupakan paket perangkat lunak yang dirancang untuk komputasi numeris dengan matriks. Pada MATLAB setiap angka dapat dinyatakan sebagai matriks sehingga lebih efisien dan mudah dibandingkan perangkat lunak lain seperti: Pascal, Fortran, Delphi, dll. Beberapa keunggulan yang dimiliki MATLAB adalah:

1. Menyediakan perintah-perintah praktis untuk menyelesaikan persoalan-persoalan matematis seperti: operasi aljabar, operasi matriks, persamaan linear dan non-linear, persamaan diferensial, korelasi data, interpolasi data, dll.
2. Memungkinkan untuk membuat fungsi-fungsi sendiri disamping beberapa fungsi yang telah disediakan (*built in*).
3. Mampu menampilkan data dalam bentuk grafik 2D ataupun 3D
4. Memiliki toolbox untuk berbagai bidang seperti: *statistics, signal processing, image processing, neural network, fuzzy logic, etc.*

5. Memiliki menu demo dan tutorial yang memungkinkan pengguna dapat belajar sendiri dengan perintah: *help*

Pada icon MATLAB double-click dan selanjutnya akan tampil Gambar 1.1 Secara umum lingkungan yang terdiri 3 bagian yaitu:

1. **Command Window**, merupakan jendela MATLAB tempat untuk mengeksekusi perintah, menampilkan masukan dan hasil.



Gambar 1.1 Command Window

Dari prompt >> pernyataan atau statemen dituliskan dan untuk mengeksekusi perintah diakhir dengan menekan <ENTER>

2. **Workspace**, berfungsi menyimpan secara otomatis segala variabel masukan dan hasil
3. **Command history**, berfungsi menyimpan secara otomatis segala perintah yang telah ditulis pada command window

Vektor dan Matriks

Vektor adalah merupakan matriks yang hanya terdiri dari satu baris atau satu kolom saja. Cara menuliskan matriks pada MATLAB dimulai dengan tutup buka ([]) dan menggunakan koma (,) atau spasi untuk memisahkan elemen-elemen satu baris, sedangkan titik koma (;) untuk memisahkan elemen-elemen satu kolom serta diakhiri dengan tutup kurung (]).

1. Menyatakan suatu matriks

Misal matriks A:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

Cara menuliskan:

```
» A = [1 2 3; 4 5 6; 7 8 9]
» A(2,1)
» B = [1:1.5:6; 2 3 4 5]
» D = []; D=[D;5]; D=[D;6;7]
» E = zeros(4, 5)
» F = ones(3, 4)
» G = rand(3,2)
```

2. Penjumlahan dan pengurangan suatu matriks

```
» b1 = [1 2 3]
» b2 = [4 5 6]
» b1 + b2
» b1 - b2
```

Jika ukuran vektor tidak sesuai maka akan mendapat *error*

```
» b = [1; 2; 3];
» c = [4 5 6];
» b+c
```

Error using ==> + Matrix dimensions must agree.

3. Perkalian

```
» b1
» b2
» b1*b2
```

4. Transpose matriks

```
» b2'
» b1*b2'
» b2'
» b1*b2'
```

Berikan komentar anda dari hasil yang diperoleh dengan menjalankan perintah diatas! Ingat aturan: Jika anda mempunyai 2 matriks dengan ukuran $r_1 \times c_1$ dan $r_2 \times c_2$, anda dapat mengalikan keduanya dengan syarat $c_1 = r_2$.

```

» A = [1 2; 3 4; 5 6]
» b = [1;2;3]
» A*b
» A'*b
» A = [1 2; 3 4; 5 6]
» b = [1;2;3]
» A*b
» A'*b

```

Perkalian elemen-elemen matriks dengan notasi '.*'

```
» b1.*b2
```

Perkalian skalar

```
» 2*c
```

5. Pembagian

Elemen per elemen :

```
» b1./b2
```

Skalar :

```
» c/2
```

6. Operator titik dua (:)

```
» x = 0:0.1:5    Menghasilkan vektor baris 51 elemen
```

7. Mensubstitusi elemen matriks

```
» A = [1 2 3; 4 5 6; 7 8 9]
```

```
» A(3,3) = 0
```

8. Operator titik dua (:) dalam matriks

```
» A(2,:) 
```

```
» A(:,2:3)
```

Menghapus baris atau kolom dengan vektor kosong []

```
» A(:,2)=[ ]
```

9. Ukuran matriks

```
» size(A)
```

```
» [m,n]=size(A)
```

10. Determinan matriks

```
» det(A)
```

11. Invers matriks

» $inv(A)$

12. Penulisan lanjutan dalam matriks

```
» G=[1 2 3 ...  
    2 2 3 ...  
    4 3 4]  
» G=[1 2 3; ...  
    2 2 3; ...  
    4 3 4]
```

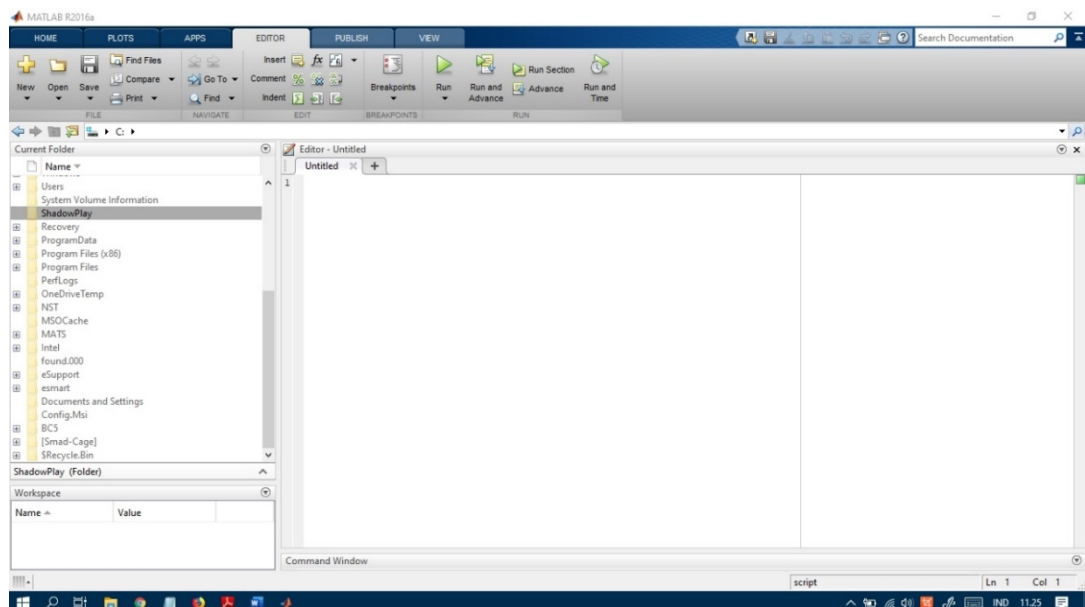
Perhatikan hasilnya!

Membuat M-File

Pada saat ini dalam memasukkan perintah masih dilakukan melalui prompt >> command window. Untuk memudahkan dalam membuat program, pada MATLAB disediakan editor atau m-file untuk mengetikkan perintah-perintah dan menyimpannya. Penulisan program m-file dapat berupa skrip program atau fungsi.

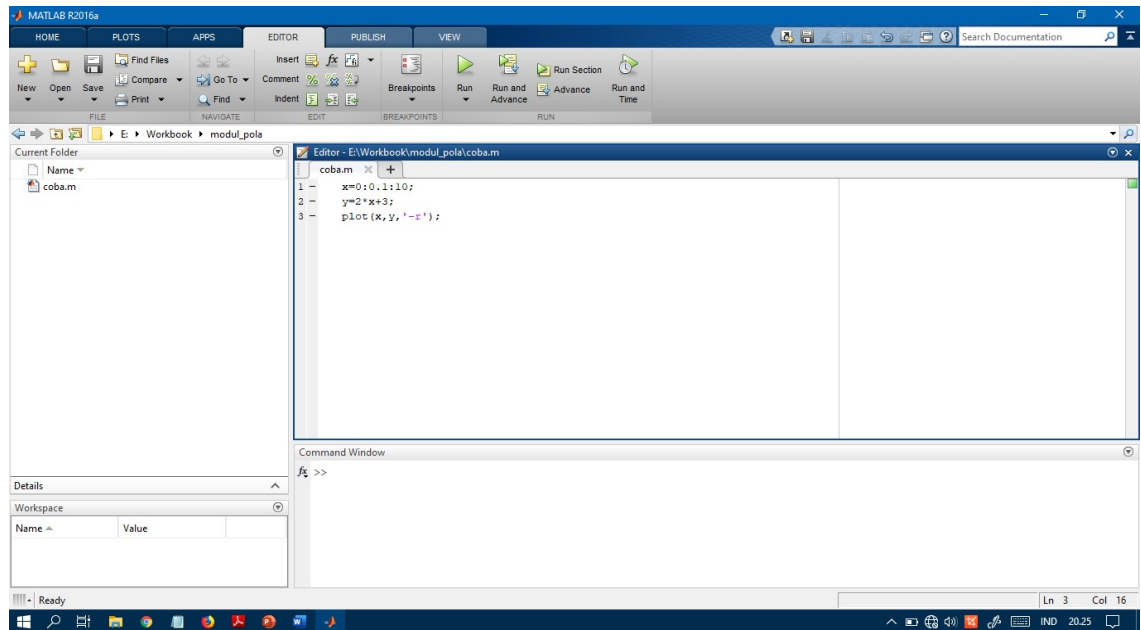
Cara membuka layar editor mulai menekan klik pada File

→ New → M-File sebagaimana pada Gambar 1.2, berikut:



Gambar 1.2 Layar Editor

Setelah di-klik pada M-File akan tampil layar editor yang siap untuk menuliskan skrip program, sebagaimana pada Gambar 1.3 berikut:



Gambar 1.3 Tampilan Layar coba.m

Skrip M-file Skrip adalah sekumpulan perintah atau pernyataan dalam suatu file yang sangat sederhana karena tidak mempunyai *input* argumen maupun *output* argument. Contoh skrip sederhana yang disimpan dengan nama file: coba.m

```
%Nama File: coba.m

%=====

clear all

x=0:0.1:10;

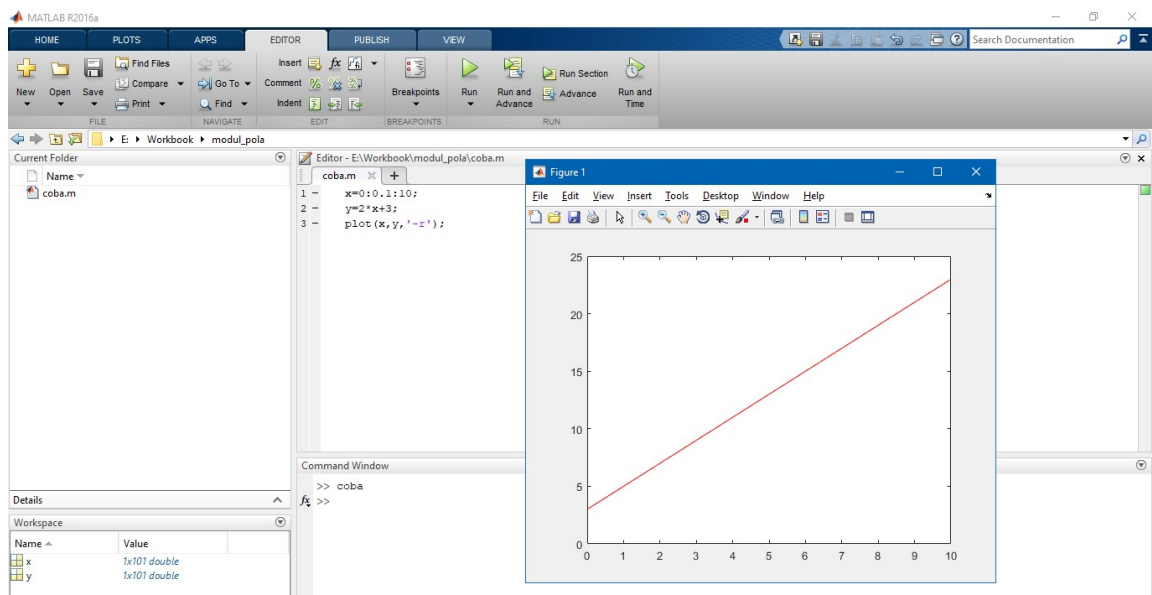
y=2*x+3;

plot(x,y,'-r')
```

Komentar pada m-file dapat dilakukan dengan menambahkan % sebelum menuliskan pernyataan. Hal ini berarti pernyataan atau perintah yang diawali % tidak di-eksekusi. Menjalankan atau eksekusi program dapat dilakukan dengan beberapa cara:

- Tekan F5 pada keyboard → Klik debug → run

- Aktifkan command window dan ketik nama file yang akan dijalankan kemudian tekan ENTER



Gambar 1. 2. Hasil output coba.m

Operasi kalang dan struktur perulangan (loop)

1. for ... end

Bentuk umum sintak nya :

```
for variabel = ekspresi
    statemen-statemen
end
```

Contoh 1:

```
for i=0:2:20
    disp(i)
end
```

Catat dan pahami hasil eksekusi-nya!

Contoh 2:

```
for i=1:4
    for j=1:3
        C(i,j)=i*j;
    end
end
```

Catat dan pahami hasil eksekusi-nya!

Contoh 3:

```
n = 5;
A = eye(n);
for j=2:n
    for i=1:j-1
        A(i,j)=i/j;
        A(j,i)=i/j;
    end
end
```

Catat dan pahami hasil eksekusi-nya!

2. while ... end

Bentuk umum sintak nya:

```
while ekspresi
    statemen-statement
end
```

Contoh 4:

```
x = 1
while x <= 10
    x = 3*x
end
```

Catat dan pahami hasil eksekusi-nya!

3. if ... end

Struktur dapat bervariasi:

- if ... end
- if ... else ... end
- if ... elseif ... else ... end

Bentuk umum sintak nya:

```
if ekspresi
    statemen-statement
end
```

Beberapa contoh didasarkan pada rumus kuadratis dengan variasi nilai diskriminan

1. $D = b^2 - 4ac$;

```
if D < 0
    disp('Peringatan: nilai diskriminan negatif, akar-akarnya
    imajiner');
end
```

2. $D = b^2 - 4ac$;

```
if D < 0
    disp(' Peringatan: nilai diskriminan negatif, akar-akarnya
    imajiner');
else
    disp('Akar-akar real, namun mungkin berbeda')
end
```

3. $D = b^2 - 4ac$;

```
if D < 0
    disp(' Peringatan: nilai diskriminan negatif, akar-akarnya
    imajiner');
elseif D == 0
    disp('Nilai diskriminan nol, akar-akar sama')
else
    disp('akar-akar riil')
end
```

1.3. ALAT DAN BAHAN

Alat dan bahan yang digunakan dalam praktikum ini yaitu:

1. Komputer.
2. Software Matlab

1.4. LANGKAH PRAKTIKUM

Misalkan dua titik p_1 dan p_2 mempunyai koordinat masing-masing:

$$p_1 = (1, 5) \quad p_2 = (4, 7)$$

kita ingin menghitung jarak antara dua tersebut. Dengan menggunakan teorema Pythagoras, maka jarak d dapat dihitung dengan rumus:

$$d = \sqrt{s_1^2 + s_2^2}$$

$$d = \sqrt{(4 - 1)^2 + (7 - 5)^2} = \sqrt{13} = 3,61$$

```
%Program menghitung jarak 2 titik

%Nama file : kasus1.m

%=====

clc

clear

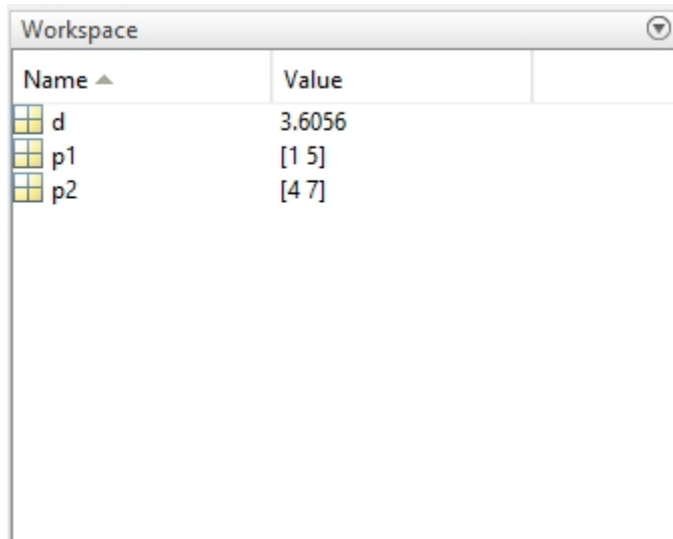
p1=[1,5];

p2=[4,7];

d=sqrt(sum((p2-p1).^2))
```

Penyelesaian dengan MATLAB dapat dibuat skrip program dengan nama file: **kasus1.m**

Jalankan program **kasus1.m** dan catat hasilnya!



Name ▲	Value
d	3.6056
p1	[1 5]
p2	[4 7]

Gambar 1. 3 Hasil skrip kasus1.m

2. Contoh kasus 2:

Sekumpulan data statistik berikut : [2.5 8.2 -1.1 -0.2 1.5]

Hitunglah (menggunakan kalkulator) :

- rerata (μ)

$$\mu = \frac{\sum_{i=1}^N x_i}{N}$$

- variansi (σ^2)

$$\sigma^2 = \frac{\sum_{i=1}^N (x_i - \mu)^2}{N - 1}$$

- standar deviasi (σ)

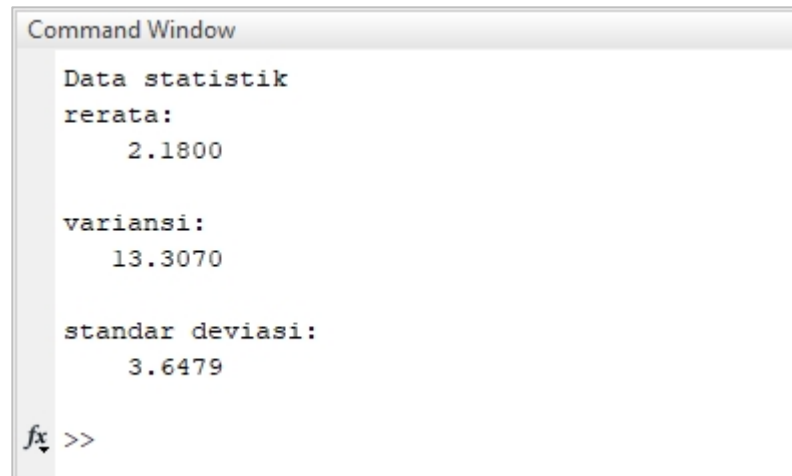
$$\sigma = \sqrt{\sigma^2}$$

Bandungkan hasilnya dengan cara membuat program MATLAB dengan menggunakan fungsi-fungsi bawaannya, nama file: **kasus2.m**

```
%Program Statistik  
  
%Nama File : kasus2.m  
  
%=====
```

```
clc  
  
clear  
  
x=[2.5 8.2 -1.1 -0.2 1.5];  
  
disp('Data statistik ');  
  
disp('rerata: ')  
  
disp(mean(x))  
  
disp('variansi: ')  
  
disp(std(x)^2)
```

```
disp('standar deviasi: ')\ndisp(std(x))
```



Gambar 1. 4. Hasil skrip kasus2.m

Jalankan program **kasus2.m** dan catat hasilnya!

3. Fungsi M-file

Fungsi adalah m-file yang memiliki *input* argumen dan *output* argumen. Beberapa bagian dari sebuah fungsi untuk menghitung volume balok adalah

```
function hasil=volume(p,l,t)
```

dimana function adalah keyword hasil adalah *output* argumen volume adalah nama fungsi p,l,t adalah *input* argumen

Nama fungsi secara default sama dengan nama variabelnya. Pada Matlab juga telah ada fungsi-fungsi yang *built in* seperti: sin(x), cos(x), exp(x), log(x), mean(x), sum(x), std(x), dll.

Contoh kasus 3:

Menghitung volume bola dengan jari-jari sebagai *input* argumen. Rumus volume bola adalah:

$$V = \frac{4}{3} \pi r^3$$

Maka untuk membuat fungsi nya adalah

```
function V=volumebola(r)
```

$$V = \frac{4}{3} \pi r^3$$

Bila fungsi diatas dijalankan dari command window, maka :

```
>> volumebola(5)
V = 523.5988
```

Argumen fungsi dapat memiliki *output* argumen lebih dari satu dan nilainya ditulis dalam sebuah kurung siku. Sedangkan untuk *input* argumen ditulis dalam kurung biasa. Pemisahan antar argumen digunakan tanda koma.

Contoh kasus 4:

Mencari akar-akar dari persamaan kuadrat, didapat rumus:

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Fungsi akar yang akan dibuat mempunyai tiga argumen *input* yaitu a,b,c dan 2 argumen *output* yaitu x1 dan x2.

Maka untuk membuat fungsi nya adalah

```
function [x1, x2]=akar2(a,b,c)
x1=(-b+sqrt(b^2-4*a*c))/2*a;
x2=(-b-sqrt(b^2-4*a*c))/2*a;
```

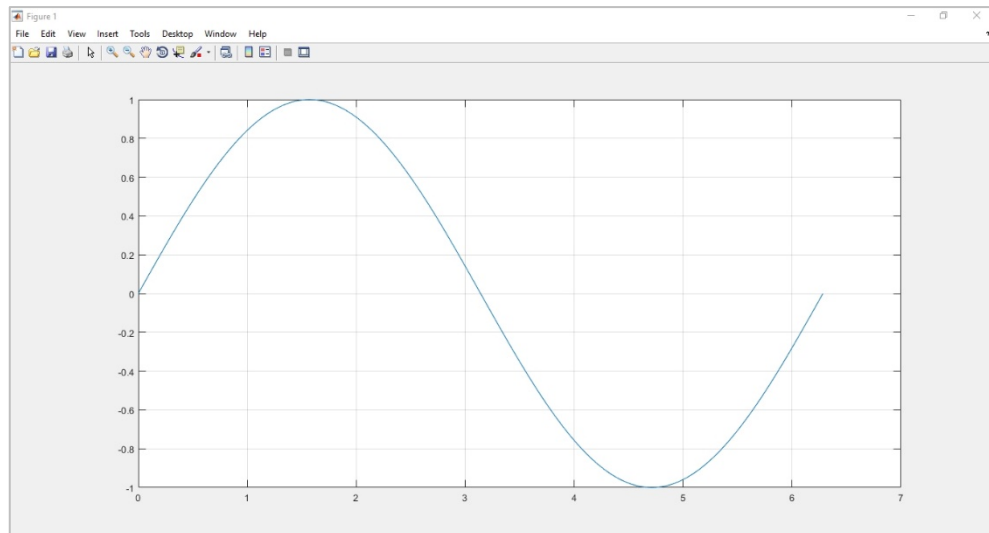
Bila fungsi diatas dijalankan dari command window, maka :

```
>> [x1,x2]=akar2(1, -1, -12)
x1 =
    4
x2 =
   -3
```

4. Menampilkan satu grafik dengan perintah plot

Contoh kasus 5:

Sebuah fungsi sinus $y = \sin(x)$ dalam interval $0 \leq x \leq 2\pi$. Buatlah m-file dengan nama **kasus5.m**, yang terdiri dari perintah-perintah berikut:



Gambar 1. 5 Hasil skrip kasus5.m

5. Menampilkan banyak grafik

Contoh kasus 6:

Plot grafik fungsi cosinus, $y_1 = 2 \cos(x)$, $y_2 = \cos(x)$, and $y_3 = 0,5 \cos(x)$, dalam interval $0 \leq x \leq 2\pi$. Buatlah m-file dengan nama kasus6.m, yang terdiri dari perintah-perintah berikut:

```
%Menampilkan Banyak Grafik

%Nama File : kasus6.m

%=====

x= 0:pi/100:2*pi;

y1 = 2*cos(x);

y2 = cos(x);

y3 = 0.5*cos(x);

plot(x,y1,'--',x,y2,'-',x,y3,':')

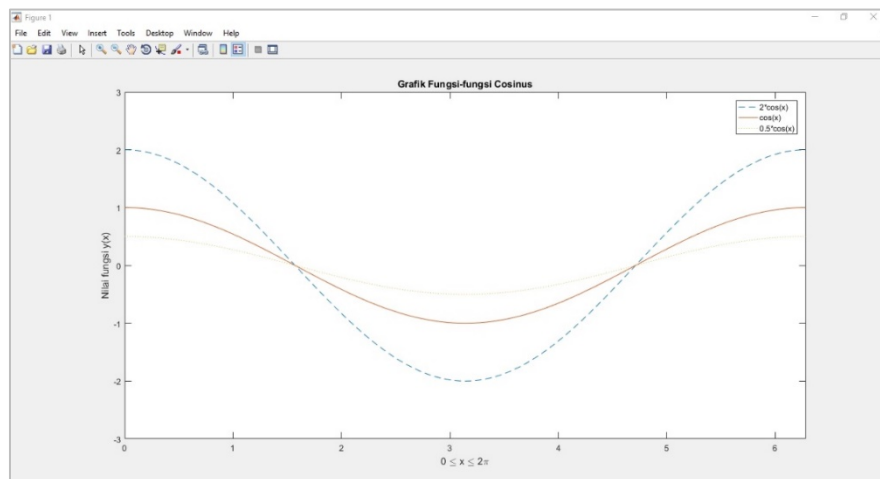
xlabel('0 \leq x \leq 2\pi')

ylabel('Nilai fungsi y(x)')

legend('2*cos(x)', 'cos(x)', '0.5*cos(x)')
```

```
title('Grafik Fungsi-fungsi Cosinus')
axis([0 2*pi -3 3])
```

Hasil Output Kode di atas



Gambar 1. 6 Hasil Output kasus6.m

- Menampilkan banyak grafik dalam satu gambar

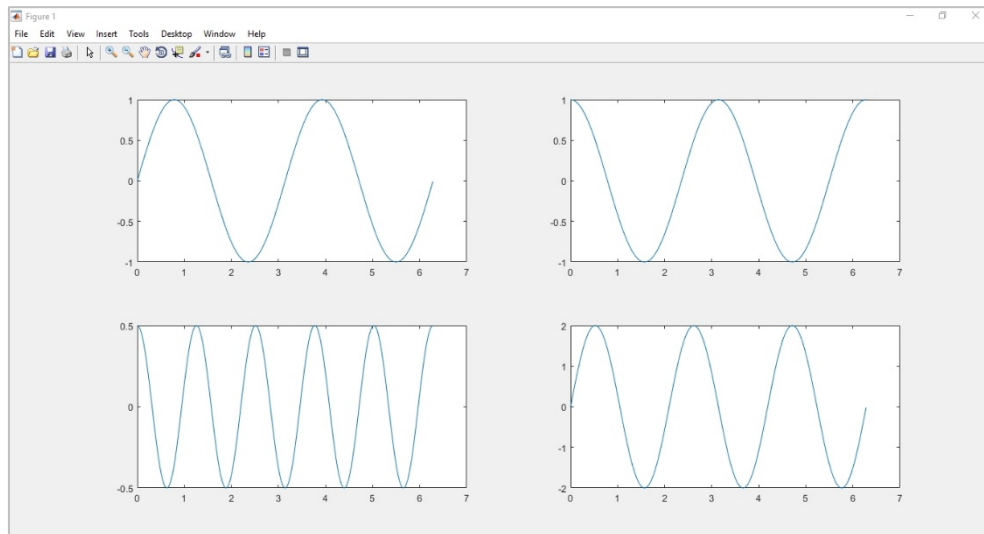
Contoh kasus 7:

Buatlah m-file dengan nama kasus7.m, yang terdiri dari perintah-perintah berikut:

```
%Menampilkan Banyak Grafik dalam Satu Gambar
%Nama File : kasus7.m
%=====
x= 0: 0.01:2*pi;
y1 = sin(2*x);
y2 = cos(2*x);
y3 = 0.5*cos(5*x);
```

```
y4 = 2*sin(3*x);  
  
subplot(2,2,1), plot(x,y1)  
  
subplot(2,2,2), plot(x,y2)  
  
subplot(2,2,3), plot(x,y3)  
  
subplot(2,2,4), plot(x,y4)
```

Hasil output kode di atas



Gambar 1. 7 Hasil skrip kasus7.m

1.5 TUGAS

Tugas berisi post test yang harus dikerjakan oleh mahasiswa sebagai evaluasi dari praktikum yang dilakukan (contoh lembar evaluasi terlampir).

LEMBAR JAWABAN PRE-TEST / POST-TEST / EVALUASI PRAKTIKUM 1: NAMA PRAKTIKUM

Nama : NIM :	Asisten: Paraf Asisten:	Tanggal: Nilai:
-------------------------------	--	----------------------------------

--

PRAKTIKUM 2: PEMROSESAN AWAL

Pertemuan ke : 2

Total Alokasi Waktu : 90 menit

- Pre-Test : 10 menit
- Praktikum : 65 menit
- Post-Test : 15 menit

Total Skor Penilaian : 100%

- Pre-Test : 25 %
 - Praktikum : 40 %
 - Post-Test : 35%
-

2.1. TUJUAN DAN INDIKATOR CAPAIAN

Setelah mengikuti praktikum ini mahasiswa diharapkan:

1. Mahasiswa mampu menjelaskan operasi dasar untuk mengolah citra

Indikator ketercapaian diukur dengan:

1. Membuat program untuk membaca, mengkonversi dan membuat histogram citra menggunakan MATLAB

2.2. TEORI PENDUKUNG

Pengenalan merupakan suatu hal yang mudah dilakukan oleh manusia, namun tidak demikian bagi sebuah mesin atau komputer. Pengenalan pola citra telah berkembang pesat seiring dengan kemajuan perangkat perangkat keras dan lunak komputer.

Beberapa contoh pengenalan citra seperti: tulisan tangan dan tandatangan, citra-citra medis, citra wajah, dll. Pada sistem pengenalan citra diawali dengan proses pengolahan citra kemudian dilanjutkan dengan ekstraksi citra dan klasifikasi pola citra

Pengolahan citra merupakan tahap awal atau pemrosesan awal dari suatu sistem pengenalan. Beberapa proses pengolahan citra yang penting dilakukan untuk pengenalan pola seperti: perubahan dari citra berwarna menjadi citra keabuan, konversi dari citra keabuan ke citra biner dengan metode pengambangan (*thresholding*). Sedangkan untuk ekstraksi ciri dapat digunakan metode histogram atau statistik dasar yang lain.

2.3. ALAT DAN BAHAN

Alat dan bahan yang digunakan dalam praktikum ini yaitu:

1. Komputer.
2. Software Matlab

2.4. LANGKAH PRAKTIKUM

2.4.1 Pembacaan citra

Pada matlab fungsi untuk melakukan pembacaan citra standar yaitu:

`imread('namafile')`

Setelah pembacaan citra dapat dilakukan proses seperti: mengubah citra berwarna RGB menjadi berderajat keabuan (*grayscale*), mengubah menjadi citra negatif dengan perintah dalam skrip berikut:

```
%Nama File: citra1.m

%=====

clear all;

close all;

I=imread('peppers.png');

figure (1)

imshow(I);

title('Citra asli');

figure (2)

G = rgb2gray(I);

imshow(G);

title('Citra keabuan');
```

Catat dan pahami fungsi bawaan Matlab **rgb2gray**

dengan eksekusi perintah

`>> help rgb2gray`

2.4.2. Membuat Konversi Citra berwarna menjadi Citra skala keabuan (*Grayscale*)

Operasi konversi citra true color ke keabuan (*grayscale*) dengan rumus sebagai berikut:

$$W_{\text{grayscale}} = \frac{(R + G + B)}{3}$$

Contoh:

```
%Nama File: citra2.m

%=====

clc

figure (1)

Image= imread('peppers.png');

imshow(Image)

red=Image(:,:,1);

green=Image(:,:,2);

blue=Image(:,:,3);

gray=(red+green+blue)/3;

figure (2)

imshow(gray)
```

Perhatikan hasil eksekusinya dan bandingkan dengan hasil melalui penggunaan fungsi **rgb2gray**

2.4.3. Membuat Citra biner dari Citra skala keabuan (*Grayscale*)

Suatu metode yang digunakan untuk mengubah citra dengan format keabuan menjadi citra biner adalah pengambangan (*thresholding*). Rumus yang digunakan untuk konversi adalah

$$Ko = \begin{cases} 0, & \text{jika } Ki < \text{ambang} \\ 1, & \text{jika } Ki \geq \text{ambang} \end{cases}$$

atau

$$Ko = \begin{cases} 0, & \text{jika } Ki \geq \text{ambang} \\ 1, & \text{jika } Ki < \text{ambang} \end{cases}$$

Contoh:

```
%Nama File: ambang.m

%=====
```

```

clear all

clc

figure (1)

citra = imread('rice.png');
imshow(citra);

[baris, kolom] = size(citra);
citra = double(citra);

for i = 1:baris
    for j = 1:kolom
        if citra(i, j) < 120
            citraThresholding(i, j) = 0;
        else
            citraThresholding(i, j) = 1;
        end
    end
end

figure (2)

imshow(citraThresholding);

```

Ulangi meng-eksekusi file **ambang.m** diatas dengan mengganti nilai ambang menjadi 100 perhatikan hasilnya, dan ulangi untuk nilai ambang yang lain: 150 Jelaskan perbandingan atas hasil ketiganya!

2.4.4. Cropping dan Menyimpan Citra

Perintah untuk melakukan cropping (pemotongan bagian tertentu dari citra), menggunakan fungsi,

Imcrop(matriks citra, titik sudut_crop);

Perintah untuk melakukan penyimpanan citra menggunakan fungsi,

Imwrite(matriks citra, namafile, format ekstensi);

Contoh:

```
%Nama File: citracrop.m

%=====

clear all

I = imread('cameraman.tif');

I2 = imcrop(I,[60 40 100 90]);

figure, imshow(I), figure, imshow(I2)

imwrite(I2,'camcrop.tif','tif')
```

Catat dan pahami hasil eksekusi-nya!

2.4.5 Membuat Histogram Citra

Perintah yang disediakan MATLAB untuk membuat histogram dari citra yaitu dengan fungsi, *imhist(matriks citra)*

Contoh:

```
%Nama File: citrahist.m

%=====

clear all

I = imread ('rice.png');

GM = im2double(I);

%penambahan derau

final_length = length(GM);

noise = 0.25*randn(256);

im_noise=GM+noise;

%histogram

figure;

subplot(2,2,1);

imhist(GM);
```

```

title('histogram');

subplot(2,2,2);

imshow(GM);

title('Citra asli');

subplot(2,2,4);

imshow(im_noise);

title('Citra + derau');

subplot(2,2,3);

imhist(im_noise);

title('histogram Citra + derau');

```

2.4.6. Membuat citra bercampur derau

```

%Nama File: citranoise.m

%=====

clear all

I = imread ('cameraman.tif');

GM = im2double(I);

%penambahan derau

final_length = length(GM);

noise = 0.25*randn(256);

im_noise=GM+noise;

imwrite(im_noise,'camnoise.tif','tif')

```

2.5. TUGAS

Tugas berisi post test yang harus dikerjakan oleh mahasiswa sebagai evaluasi dari praktikum yang dilakukan (contoh lembar evaluasi terlampir).

LEMBAR JAWABAN PRE-TEST / POST-TEST / EVALUASI PRAKTIKUM 1: NAMA PRAKTIKUM

Nama : NIM :	Asisten: Paraf Asisten:	Tanggal: Nilai:
-------------------------------	--	----------------------------------

--

PRAKTIKUM 3: EKSTRAKSI CIRI

Pertemuan ke : 3

Total Alokasi Waktu : 90 menit

- Pre-Test : 10 menit
- Praktikum : 65 menit
- Post-Test : 15 menit

Total Skor Penilaian : 100%

- Pre-Test : 25 %
 - Praktikum : 40 %
 - Post-Test : 35%
-

3.1. TUJUAN DAN INDIKATOR CAPAIAN

Setelah mengikuti praktikum ini mahasiswa diharapkan:

1. Memahami tekstur citra
2. Memahami proses ekstraksi ciri
3. Membuat pola citra

Indikator ketercapaian diukur dengan: (sesuaikan dengan RPS)

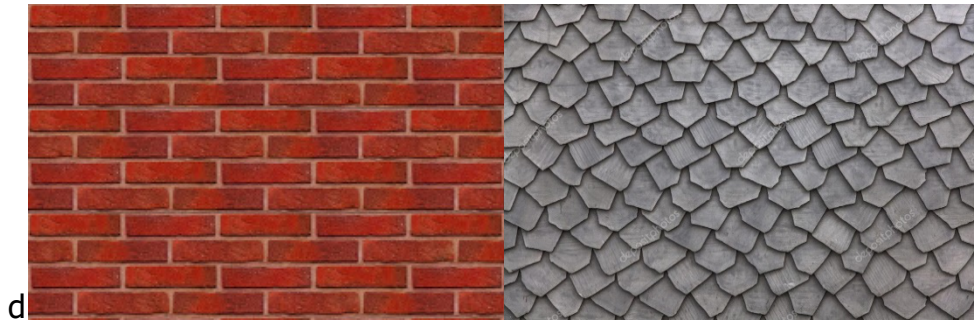
1. Membuat program sederhana menggunakan MATLAB dengan *.m file
2. Memahami statemen operasi kalang dan mampu membuat grafik.

3.2. TEORI PENDUKUNG

Tekstur merupakan karakteristik intrinsik dari suatu citra yang terkait dengan tingkat kekasaran (*roughness*), granularitas (*granulation*), dan keteraturan (*regularity*) susunan struktural piksel. Aspek tekstural dari sebuah citra dapat dimanfaatkan sebagai dasar dari segmentasi, klasifikasi, maupun interpretasi citra.

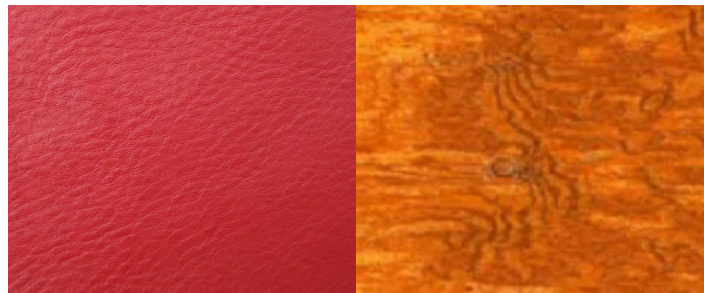
Tekstur dapat didefinisikan sebagai fungsi dari variasi spasial intensitas piksel (nilai keabuan) dalam citra. Berdasarkan strukturnya, tekstur dapat diklasifikasikan dalam dua golongan : makrostruktur dan mikrostruktur

Tekstur makrostruktur memiliki perulangan pola lokal secara periodik pada suatu daerah citra, biasanya terdapat pada pola-pola buatan manusia dan cenderung mudah untuk direpresentasikan secara matematis. Contoh tekstur makrostruktur sebagaimana pada Gambar 4, berikut:



Gambar 3. 1 Citra tesktur makrosutruktur

Pada tekstur mikrostruktur, pola-pola lokal dan perulangan tidak terjadi begitu jelas, sehingga tidak mudah untuk memberikan definisi tekstur yang komprehensif. Gambar 5 berikut ini menunjukkan contoh tekstur mikrostruktur.



Gambar 3. 2 citra teksutr mikrostruktur

Apa itu Ekstraksi Ciri?

- Untuk mengenali objek dalam suatu citra maka dibutuhkan parameter-parameter yang dapat mencirikan objek tersebut. Ekstraksi Ciri merupakan proses pengambilan ciri atau karakteristik objek yang dapat digunakan sebagai pembeda dari objek-objek lainnya.
- Sedangkan ciri warna digunakan apabila objek-objek yang akan dikenali memiliki warna yang berbeda seperti untuk membedakan citra buah apel Amerika yang berwarna merah dengan apel Malang berwarna hijau maka digunakanlah ciri warna sebagai parameternya

Ekstraksi ciri orde pertama

Ekstraksi ciri orde pertama merupakan metode pengambilan ciri yang didasarkan pada karakteristik histogram citra. Histogram menunjukkan probabilitas kemunculan nilai derajat keabuan piksel pada suatu citra. Dari nilai-nilai pada histogram yang dihasilkan, dapat dihitung beberapa parameter ciri orde pertama, antara lain adalah **mean**, **skewness**, **variance**, **kurtosis**, dan **entropy**.

a. Mean (μ)

Menunjukkan ukuran dispersi dari suatu citra

$$\mu = \sum f(x, y) \cdot p(f(x, y))$$

dimana $(f(x, y))$ merupakan suatu nilai intensitas keabuan, sementara $p(f(x, y))$ menunjukkan nilai histogramnya (probabilitas kemunculan intensitas tersebut pada citra).

b. Variance (σ^2)

Menunjukkan variasi elemen pada histogram dari suatu citra

$$\sigma^2 = \sum (f(x, y) - \mu)^2 \cdot p(f(x, y))$$

c. Skewness (α_3)

Menunjukkan tingkat kemencengan relatif kurva histogram dari suatu citra

$$\alpha_3 = \frac{1}{\sigma^3} \sum (f(x, y) - \mu)^3 \cdot p(f(x, y))$$

d. Kurtosis (α_4)

Menunjukkan tingkat keruncingan relatif kurva histogram dari suatu citra

$$\alpha_4 = \frac{1}{\sigma^4} \sum (f(x, y) - \mu)^4 \cdot p(f(x, y)) - 3$$

e. Entropy (H)

Menunjukkan ukuran ketidakaturan bentuk dari suatu citra

$$H = - \sum p(f(x, y)) \cdot \log_2(p(x, y))$$

3.3. ALAT DAN BAHAN

Alat dan bahan yang digunakan dalam praktikum ini yaitu:

1. Komputer.
2. Software Matlab

3.4. LANGKAH PRAKTIKUM

1. Buatlah fungsi **forde1.m** yang dipergunakan untuk menghitung ciri orde satu dari sebuah citra:

```

function [m,v,s,k,e]=forde1(image)

H=imhist(image)';
H=H/sum(H);
I=[0:255];

m = I*H';
v = (I-m).^2*H';
s = (I-m).^3*H'/v^1.5;
k = (I-m).^4*H'/v^2-3;
e = -H*log2(H+eps)';

```

2. Sediakan database citra tekstur makro struktur dan mikro struktur masing-masing 15 citra.
3. Ubahlah citra berwarna dari database menjadi format derajat keabuan dengan menggunakan fungsi **rgb2gray**
4. Tentukan ciri citra dengan menggunakan fungsi **forde1** . Catat hasilnya dan berikan penjelasan dari hasil yang diperoleh.
5. Dengan melakukan seleksi ciri yaitu mean dan variance, gambarkan grafik mean Vs variance
6. Jelaskan apakah dua ciri yang dipilih telah mampu membedakan antara makro struktur dan mikrostruktur.

3.5. TUGAS

Tugas berisi post test yang harus dikerjakan oleh mahasiswa sebagai evaluasi dari praktikum yang dilakukan (contoh lembar evaluasi terlampir).

Ulangi point 3-6 dengan menggunakan fungsi **warna2abu.m** yang telah dibuat pada praktikum 2.

<p>Nilai</p>	<p>Yogyakarta,</p> <p>.....</p> <p>Paraf asisten</p> <p><.....></p>

PRAKTIKUM 4: KLASIFIKASI DENGAN GARIS LINEAR

Pertemuan ke : 4

Total Alokasi Waktu : 90 menit

- Pre-Test : 10 menit
- Praktikum : 65 menit
- Post-Test : 15 menit

Total Skor Penilaian : 100%

- Pre-Test : 25 %
 - Praktikum : 40 %
 - Post-Test : 35%
-

4.1. TUJUAN DAN INDIKATOR CAPAIAN

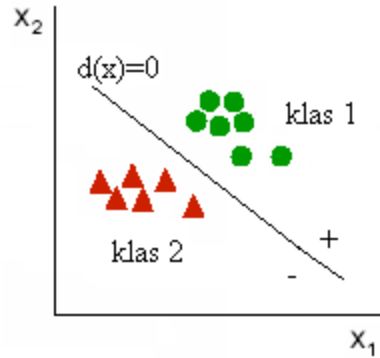
Setelah mengikuti praktikum ini mahasiswa diharapkan: Mahasiswa mampu memahami estimasi linier dan teknik klasifikasi sederhana

Indikator ketercapaian diukur dengan: (sesuaikan dengan RPS)

1. Memahami estimasi linear
2. Memahami pengklasifian garis linear

4.2. TEORI PENDUKUNG

Fungsi utama dari suatu sistem pengenalan pola adalah mengambil keputusan untuk menentukan anggota klas dari suatu pola-pola masukan. Olehkarenanya untuk melakukan tugas ini diperlukan beberapa aturan yang berdasarkan pada fungsi keputusan. Satu contoh sederhana fungsi keputusan untuk memisahkan populasi dua pola dapat digunakan persamaan garis linear, sebagaimana ditunjukkan pada Gambar 6 dibawah,



Gambar 4. 1 Fungsi keputusan untuk mengklasifikasi dua jenis pola

Persamaan garis linear, misalnya $d(x) = w_1x_1 + w_2x_2 + w_3 = 0$, dimana w adalah parameter-parameter dan x_1, x_2 adalah variabel-variabel koordinat. Dari gambar 6 diatas, jelas bahwa pola-pola $P = (x_1, x_2)$ pada klas 1 jika nilai-nilai variabel x nya disubstitusikan kedalam persamaan $d(x)$ maka hasil bernilai positif. Dengan cara yang sama $d(x)$ akan bernilai negatif jika pola-pola $P = (x_1, x_2)$ pada klas 2 disubstitusikan kedalam persamaan $d(x)$ tersebut. Olehkarena itu $d(x)$ dapat digunakan sebagai fungsi keputusan, jika misalnya diberikan suatu pola x yang belum diketahui kelompok klas nya, maka dapat ditentukan bahwa pola x akan masuk dalam kategori klas 1 jika $d(x) > 0$, atau masuk kedalam kategori klas 2 jika $d(x) < 0$.

4.3. ALAT DAN BAHAN

Alat dan bahan yang digunakan dalam praktikum ini yaitu:

1. Komputer.
2. Software Matlab

4.4. LANGKAH PRAKTIKUM

1. Jalankan program `tekstur.m` dibawah ini:
2. Catat hasilnya dan tentukan persamaan garis untuk memisahkan antara klas citra makro struktur dan citra mikro struktur berdasarkan ciri rerata (*mean*) dan variansi.
3. Modifikasi program ***tekstur.m*** diatas dan tentukan persamaan garis untuk memisahkan antara klas citra makro struktur dan citra mikro struktur berdasarkan ciri rerata (*mean*) dan *skewness*.

```
%=====tekstur.m=====
clc
clg
clear all
for i = 1:15
    citra1{i} = imread(['D: \datacitra\mikros\' num2str(i) '.jpg']);
```

```

        citra1{i}= rgb2gray(citra1{i});
        citra2{i} = imread(['D: \datacitra\makros\' num2str(i) '.jpg']);
        citra2{i}= rgb2gray(citra2{i});
    end

    rerata1=[ ];
    var1=[ ];

    rerata2=[ ];
    var2=[ ];

    for i = 1:15
        [m,v,s,k,e]=forde1(citra1{i});
        rerata1(i)=m;
        var1(i)=v;
        [rerata1]=[rerata1 rerata1(i)];
        [var1]=[var1 var1(i)];

        [m,v,s,k,e]=forde1(citra2{i});
        rerata2(i)=m;
        var2(i)=v;
        [rerata2]=[rerata2 rerata2(i)];
        [var2]=[var2 var2(i)];
    end

    plot(var1,rerata1,'*b')
    hold on
    plot(var2,rerata2,'or')
    xlabel('variansi')
    ylabel('rerata')

```

```
title('plot ekstraksi ciri')  
hold off
```

4.5. TUGAS

Ulangi untuk ciri rerata (mean) dan kurtosis serta ulangi lagi untuk ciri rerata (mean) dan entrop

Nilai	Yogyakarta, Paraf asisten <.....>

LEMBAR JAWABAN PRE-TEST / POST-TEST / EVALUASI PRAKTIKUM 1: NAMA PRAKTIKUM

Nama : NIM :	Asisten: Paraf Asisten:	Tanggal: Nilai:
-------------------------------	--	----------------------------------

--

PRAKTIKUM 5: KLASIFIKASI DENGAN METRIK JARAK

Pertemuan ke	: 5
Total Alokasi Waktu	: 90 menit
• Pre-Test	: 10 menit
• Praktikum	: 65 menit
• Post-Test	: 15 menit
Total Skor Penilaian	: 100%
• Pre-Test	: 25 %
• Praktikum	: 40 %
• Post-Test	: 35%

5.1. TUJUAN DAN INDIKATOR CAPAIAN

Setelah mengikuti praktikum ini mahasiswa diharapkan:

2. Mahasiswa mampu menjelaskan operasi dasar menggunakan Matlab
3.
4. Dst.

Indikator ketercapaian diukur dengan:

1. Memahami proses klasifikasi pola
2. Memahami pengklasifikasi metrik jarak

5.2. TEORI PENDUKUNG

Satu ide dasar penggunaan fungsi jarak sebagai alat pengklasifikasi adalah kenyataan bahwa kemiripan atau perbedaan antara pola satu dengan dengan pola-pola yang lain dapat telah terkuantisasi dapat diukur nilai kemiripannya. Maka pengukuran kemiripan atau ketidakmiripan merupakan suatu dasar dalam tugas-tugas klasifikasi dan pengenalan. Salah satu ukuran kemiripan adalah dengan menentukan metrik jarak.

Ada beberapa metrik jarak yang cukup populer dan sering digunakan dalam sistem pengenalan pola yaitu: **Manhattan**, **Euclidean**, **Canberra**, dll.

Jika x dan y adalah dua vektor ciri d -dimensi (x = referensi/template, y = uji/test) maka metrik jarak dapat didefinisikan:

1. Metrik L1 (*Manhattan*):

$$d_M(x, y) = \sum_{i=1}^d |x_i - y_i|$$

2. Metrik L2 (*Euclidean*):

$$d_E(x, y) = \sqrt{\sum_{i=1}^d (x_i - y_i)^2}$$

3. *Canberra*:

$$d_C(x, y) = \sum_{i=1}^d \frac{|x_i - y_i|}{|x_i| + |y_i|}$$

Contoh kasus:

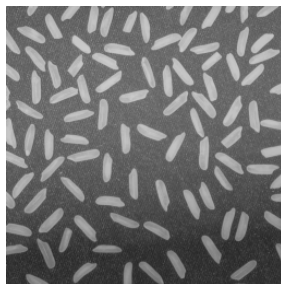
Suatu sistem pengenalan akan dilakukan klasifikasi dengan mengukur kemiripan suatu pola citra. Adapun citra referensi terdiri dari dua kelas yaitu *rice* dan *cameraman*, dan beberapa citra yang tercampur *noise*, untuk pengujian kemiripannya sebagai berikut:



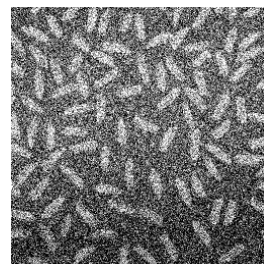
a). Gb. Cameraman



b). Gb. Cameraman ber-noise



c). Gb. Rice



d). Gb. Rice ber-noise

5.3. ALAT DAN BAHAN

Alat dan bahan yang digunakan dalam praktikum ini yaitu:

1. Komputer.
2. Software Matlab

5.4. LANGKAH PRAKTIKUM

1. Buatlah fungsi untuk menghitung jarak dengan metode Manhattan.

```
function d=dmanhattan(x,y)

    d=sum(abs(x-y));

end
```

2. Buatlah program **kenalcitra.m**, dibawah ini:

```
%Nama File: kenalcitra.m

%=====

clc;

clear;

I1=imread('rice.png');

I2=imread('cameraman.tif');

Im1=imresize(I1,[20 20],'bilinear');

x1=reshape(Im1,[400,1,3]);

Im2=imresize(I2,[20 20],'bilinear');

x2=reshape(Im2,[400,1,3]);


%test

I=imread('ricenoise.tif');

Im=imresize(I,[20 20],'bilinear');
```

```
y=reshape(Im,[400,1,3]);
```

```
d1=dmanhattan(x1,y)
```

```
d2=dmanhattan(x2,y)
```

```
minimum=min([d1 d2]);
```

```
if (d1==minimum)
```

```
    class='rice'
```

```
elseif (d2==minimum)
```

```
    class='cameraman'
```

```
end
```

Jalankan program **kenalcitra.m** diatas dan catat serta perhatikan hasilnya!

3. Modifikasi program **kenalcitra.m** dengan menggunakan database datacitra mikro struktur dan makro struktur, sebagai citra referensi masing-masing data nomer 1.
4. Buatlah pengujian dengan menggunakan citra makro dan mikro struktur no.2 -11, untuk mendapatkan unjuk kerja sistem dengan mengisi tabel berikut:

Masukan (citra)	Fungsi Jarak Manhattan (L1)	
	Mikro struktur	Makro struktur
Mikro struktur		
Makro struktur		
Akurasi (%) :		

5.5. TUGAS

Buatlah fungsi **distL2.m** untuk menghitung jarak dengan metode Euclidean. Buatlah pengujian dengan menggunakan citra makro dan mikro struktur no.2 -11, untuk mendapatkan unjuk kerja sistem dengan mengisi tabel diatas:

LEMBAR JAWABAN PRE-TEST / POST-TEST / EVALUASI PRAKTIKUM 1: NAMA PRAKTIKUM

Nama : NIM :	Asisten: Paraf Asisten:	Tanggal: Nilai:
-------------------------------	--	----------------------------------

--

PRAKTIKUM 6: SISTEM PENGENALAN WAJAH

Pertemuan ke	: 6
Total Alokasi Waktu	: 90 menit
• Pre-Test	: 10 menit
• Praktikum	: 65 menit
• Post-Test	: 15 menit
Total Skor Penilaian	: 100%
• Pre-Test	: 25 %
• Praktikum	: 40 %
• Post-Test	: 35%

6.1 TUJUAN DAN INDIKATOR CAPAIAN

Setelah mengikuti praktikum ini mahasiswa diharapkan: Mahasiswa mampu memahami teknik klasifikasi menggunakan metrik jarak

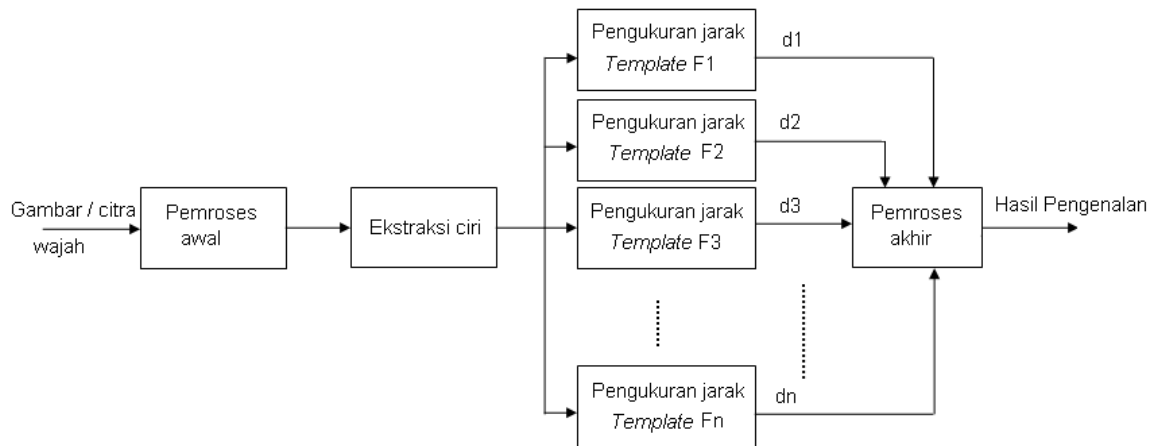
Indikator ketercapaian diukur dengan:

1. Memahami proses pengenalan wajah sederhana
2. Memahami pengklasifikasi metrik jarak

6.2 TEORI PENDUKUNG

Wajah merupakan salah satu karakteristik biometrik yang digunakan untuk mengenali seseorang selain karakteristik yang lain seperti ucapan, sidik jari, retina, dll. Praktikum ini memfokuskan pada perbandingan efektifitas penggunaan fungsi jarak dengan sebagai pengklasifikasi pada sistem biometrik pengenalan wajah. Ketiga fungsi jarak yang dikaji yaitu: Manhattan (L1), Euclidean (L2) dan Canberra.

Secara umum sistem pengenalan wajah dapat ditunjukkan pada Gambar 6, berikut:



Gambar 6. 1. Blok Diagram Sistem Pengenalan Wajah

Pada prinsipnya sistem pengenalan wajah meliputi 5 bagian yaitu: akuisisi data, pemroses awal, ekstraksi ciri, pengklasifikasi dan pemroses akhir.

Akuisisi data dan Database

Proses awal dari pengenalan wajah adalah akuisisi data yaitu pengambilan gambar wajah menggunakan kamera digital atau *webcam*. Pada praktikum ini data yang digunakan diperoleh melalui download dari: http://www.ee.surrey.ac.uk/Personal/T.Windeatt/msc_projects/. Data terbagi dalam 2 bagian yaitu data untuk pelatihan (*training set*) dan data untuk pengujian (*testing set*), dari 5 gambar wajah orang dengan notasi: F1, F2,dan F5 dalam berbagai ekspresi. Masing-masing wajah terdiri 8 sampel data. Data nomor 1-4 untuk pelatihan (*training*) sedangkan data nomor 5-8 untuk pengujian (*testing*).

Pemroses awal dan ekstraksi ciri

Pemroses awal pada dasarnya bertujuan untuk mendapatkan data dengan ukuran yang lebih kecil namun cukup mewakili data asli yang sebenarnya. Pada data gambar wajah yang digunakan dalam penelitian ini akan dilakukan penurunan resolusi gambar asli yaitu dari 112×92 piksel menjadi 30×20 piksel. Selanjutnya gambar wajah yang juga dapat dikatakan sebagai matriks berukuran 30×20 tersebut diubah menjadi suatu bentuk pola berupa matriks berukuran 600×1.

Ekstraksi ciri bertujuan untuk menajamkan perbedaan-perbedaan pola sehingga akan sangat memudahkan dalam pemisahan kategori kelas pada proses klasifikasi. Gambar/citra wajah setelah melalui pemroses awal dapat dinyatakan dalam bentuk pola :

$$Im = [\Gamma_1, \Gamma_2, \dots, \Gamma_n]$$

Rerata pola citra Ψ dan pola dengan pengurangan reratanya Φ , didefinisikan dengan:

$$\psi = \frac{1}{M} \sum_{n=1}^M \Gamma_n$$

$$\Phi_i = \Gamma_i - \psi$$

Φ_i merupakan pola hasil ekstraksi ciri yang akan digunakan sebagai masukan pada proses pengklasifikasi.

Pengklasifikasi dan Pemroses Akhir

Satu ide dasar penggunaan fungsi jarak sebagai alat pengklasifikasi adalah kenyataan bahwa kemiripan atau perbedaan antara pola satu dengan pola yang lain dapat diukur nilai kemiripannya. Maka pengukuran kemiripan atau ketidakmiripan merupakan suatu dasar dalam tugas-tugas klasifikasi dan pengenalan. Pada sistem pengenalan wajah yang dibuat ini akan dibandingkan efektifitas dua fungsi jarak yaitu Manhattan (L1), Euclidean (L2) dan Canberra, sebagai pengklasifikasi. Ketiga fungsi jarak tersebut sebagaimana terdapat pada praktikum sebelumnya.

Pola template merupakan pola yang akan digunakan sebagai referensi pada proses pengukuran jarak. Pola ini didasarkan pada data pelatihan (*training set*) yang digunakan dengan menghitung rerata dari sejumlah N data pelatihan. Pola template wajah untuk masing-masing orang dapat diperoleh dari:

$$\overline{\Phi}_i^k = \frac{1}{N} \sum_{k=1}^N \Phi_i^k$$

Pemroses akhir merupakan suatu proses tahap terakhir untuk pengambilan keputusan hasil pengukuran jarak atau tingkat kemiripan. Suatu pola baru yang belum dikenal oleh sistem dapat dikatakan mirip dengan salah satu pola template/referensi jika telah dilakukan proses penghitungan nilai jarak antara pola baru tersebut dengan setiap pola template/referensi. Kategori kemiripan didasarkan pada nilai jarak minimum, yang dapat didefinisikan:

$$k^* = \arg \min_k d_k, \quad 1 \leq k \leq n$$

dimana, n adalah jumlah orang yang akan dikenali.

6.3 ALAT DAN BAHAN

Alat dan bahan yang digunakan dalam praktikum ini yaitu:

1. Komputer.
2. Software Matlab

6.4 LANGKAH PRAKTIKUM

1. Buatlah program **KenalWajah.m**

```
%=====
%Nama file : KenalWajah.m
%=====
% Membaca data wajah untuk pelatihan dari 5 orang yang berbeda
% dengan masing-masing menggunakan 4 sampel data
clc;
for i = 1:4
    data1{i} = imread(['C:\MATLAB7\work\face\f1\' num2str(i) '.tif']);
    data2{i} = imread(['C:\MATLAB7\work\face\f2\' num2str(i) '.tif']);
    data3{i} = imread(['C:\MATLAB7\work\face\f3\' num2str(i) '.tif']);
    data4{i} = imread(['C:\MATLAB7\work\face\f4\' num2str(i) '.tif']);
    data5{i} = imread(['C:\MATLAB7\work\face\f5\' num2str(i) '.tif']);
end;
% Pemrosesan awal dan ekstraksi ciri
for i = 1:4
    data1{i} = imresize(data1{i},[30 20]);
    x1{i}=reshape(data1{i},[600,1]);
    x1{i}=x1{i}-mean(x1{i});
    x1{i}=
    data2{i} = imresize(data2{i},[30 20]);
    x2{i}=reshape(data2{i},[600,1]);
    x2{i}=x2{i}-mean(x2{i});
    data3{i} = imresize(data3{i},[30 20]);
    x3{i}=reshape(data3{i},[600,1]);
    x3{i}=x3{i}-mean(x3{i});
    data4{i} = imresize(data4{i},[30 20]);
    x4{i}=reshape(data4{i},[600,1]);
    x4{i}=x4{i}-mean(x4{i});
    data5{i} = imresize(data5{i},[30 20]);
    x5{i}=reshape(data5{i},[600,1]);
```

```

        x5{i}=x5{i}-mean(x5{i});
end;
% Template 5 wajah ('training mode')
p1=mean([x1{1} x1{2} x1{3} x1{4}],2);
p2=mean([x2{1} x2{2} x2{3} x2{4}],2);
p3=mean([x3{1} x3{2} x3{3} x3{4}],2);
p4=mean([x4{1} x4{2} x4{3} x4{4}],2);
p5=mean([x5{1} x5{2} x5{3} x5{4}],2);
%'testing mode'
I=imread(['C:\MATLAB7\work\face\f1\5.tif']);
x= imresize(I,[30 20]);
x=reshape(x,[600,1]);
px=double(x-mean(x));
d1=distL1(px,p1);
d2=distL1(px,p2);
d3=distL1(px,p3);
d4=distL1(px,p4);
d5=distL1(px,p5);
%Pengambilan keputusan hasil pengenalan
minimum=min([d1 d2 d3 d4 d5]);
if (d1==minimum)
    class='f1'
elseif (d2==minimum)
    class='f2'
elseif (d3==minimum)
    class='f3'
elseif (d4==minimum)
    class='f4'
else (d5==minimum)
    class='f5'
end

```

2. Jalankan dan Catat hasil pengamatannya, ulangi dengan mengganti masukan file citra yang diujikan dari masing-masing orang (F1-F4) pada nomor citra 5-8. Sajikan hasilnya dengan mengisi tabel berikut:

Masukan	Fungsi Jarak Manhattan (L1)				
	F1	F2	F3	F4	F5
F1					
F2					
F3					
F4					
F5					
Akurasi (%)					

6.5 TUGAS

1. Buatlah fungsi **distL2.m** untuk menghitung jarak dengan metode Euclidean. Kerjakan seperti pada point 3 dengan menggunakan metode Euclidean dan sajikan hasilnya dengan mengisi tabel serupa diatas.
2. Buatlah fungsi **canberra.m** untuk menghitung jarak dengan metode Canberra. Kerjakan seperti pada point 3 dengan menggunakan metode Canberra dan sajikan hasilnya dengan mengisi tabel serupa diatas.

Nilai	Yogyakarta, Paraf asisten <.....>

LEMBAR JAWABAN PRE-TEST / POST-TEST / EVALUASI PRAKTIKUM 1: NAMA PRAKTIKUM

Nama : NIM :	Asisten: Paraf Asisten:	Tanggal: Nilai:
-------------------------------	--	----------------------------------

--

PRAKTIKUM 7: JARINGAN SYARAF TIRUAN UNTUK PENGENALAN POLA

Pertemuan ke : 7

Total Alokasi Waktu : 90 menit

- Pre-Test : 10 menit
- Praktikum : 65 menit
- Post-Test : 15 menit

Total Skor Penilaian : 100%

- Pre-Test : 25 %
 - Praktikum : 40 %
 - Post-Test : 35%
-

7.1. TUJUAN DAN INDIKATOR CAPAIAN

Setelah mengikuti praktikum ini mahasiswa diharapkan:

Mahasiswa mampu memahami algoritma hebb pada jaringan syaraf tiruan

Indikator ketercapaian diukur dengan:

1. Memahami konsep jaringan syaraf tiruan
2. Memahami algoritma hebb
3. Memahami aplikasi algoritma hebb pengenalan pola

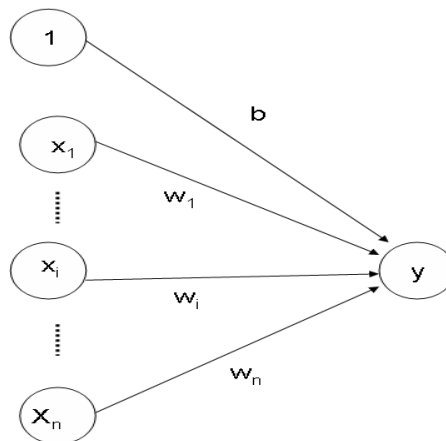
7.2. TEORI PENDUKUNG

Jaringan saraf tiruan (JST) merupakan sistem pemroses informasi yang unjuk kerjanya didasarkan pada jaringan syaraf biologis, yangtelah dikembangkan dengan beberapa asumsi-asumsi: proses informasi terjadi pada banyak elemen sederhana yang disebut neuron, sinyal antara neuron lewat link koneksi yang mana setiap link koneksi mempunyai bobot. Setiap neuron mempunyai fungsi aktivasi input yang merupakan jumlah bobot sinyal masukan untuk menentukan sinyal output.

Karakteristik JST terdiri dari:

1. Model koneksi antar neuron (arsitektur)
2. Metode penentuan bobot koneksi (algoritma pelatihan)
3. Fungsi aktivasi

Arsitektur:



Gambar 7. 1. Arsitektur Jaringan Syaraf Tiruan Sederhana

Algoritma Pembelajaran Hebb

Pelatihan dengan memodifikasi kekuatan sinapsis (bobot) yaitu bila 2 neuron yang terhubung dan keduanya pada kondisi 'hidup' pada saat yang sama, maka bobot tersebut harus ditambah. Apabila data disajikan secara bipolar, maka perbaruan bobotnya adalah:

$$w_i(\text{baru}) = w_i(\text{lama}) + x_i y$$

Algoritma:

L0. Inisialisasi semua bobot:

$$w_i = 0; \quad \text{dengan } i = 1, 2, \dots, n$$

L1. Untuk setiap pasangan vektor pembelajaran input-output lakukan langkah 2 - 4.

L2. Tetapkan aktivasi unit input:

$$x_i = s_i; \quad \text{dengan } i = 1, 2, \dots, n$$

L3. Tetapkan aktivasi unit output:

$$y = t;$$

L4. Atur bobot:

$$w_i(\text{baru}) = w_i(\text{lama}) + x_i y \quad \text{dengan } i = 1, 2, \dots, n$$

Atur bias:

$$b(\text{baru}) = b(\text{lama}) + y$$

Fungsi aktivasi

Fungsi aktivasi adalah salah satu parameter yang terpenting dalam jaringan syaraf tiruan. fungsi ini tidak hanya menentukan keputusan garis, disamping nilai fungsi aktivasi juga menunjukkan total signal

dari node. Oleh karena pemilihan fungsi aktivasi tidak dapat secara sembarangan dipilih sebab sangat besar berdampak pada performan jaringan syaraf tiruan.

Beberapa fungsi aktivasi dirumuskan sebagai:

- a. Fungsi Bipolar

$$f(x_j) = \begin{cases} 1 & x_j \geq 0 \\ -1 & x_j < 0 \end{cases}$$

- b. Fungsi Saturating Linear

$$f(x_j) = \begin{cases} 0 & x_j \leq 0 \\ x_j & 0 < x_j < 1 \\ 1 & x_x \geq 1 \end{cases}$$

- c. Fungsi Sigmoid

$$f(x_j) = \frac{1}{1 + e^{-\lambda_j x_j}}$$

7.3. ALAT DAN BAHAN

Alat dan bahan yang digunakan dalam praktikum ini yaitu:

1. Komputer.
2. Software Matlab

7.4. LANGKAH PRAKTIKUM

1. Memahami fungsi Aktivasi

- a. Fungsi bipolar dengan threshold (θ)

$$y = f(x) = \begin{cases} 1; & \text{jika } x > \theta \\ 0; & \text{jika } -\theta \leq x \leq \theta \\ -1; & \text{jika } x < -\theta \end{cases}$$

Buatlah fungsi **bipolar2.m** file dari fungsi di atas

```
function y=bipolar2(x,th)
    if x > th
        y=1;
```

```

elseif x < -1*th
    y=-1;
else
    y=0;
end
end
end

```

Catatlah hasil nya dengan menggunakan fungsi *bipolar2.m* dengan threshold = 0.5 untuk:

- i. $f(0.8, 0.5) = \dots$ iii. $f(-0.2, 0.5) = \dots$
- ii. $f(35, 0.5) = \dots$ iv. $f(-4.5, 0.5) = \dots$

b.. Fungsi Sigmoid

$$y = f(x) = \frac{1}{1 + e^{-ax}}$$

Buatlah fungsi **sigmoid.m** file dari fungsi diatas

```

function y=sigmoid(x)

    a=0.5;

    y=1./(1+exp(-1*a.*x))

end

```

Catatlah hasil nya dengan menggunakan fungsi *sigmoid.m* untuk:

- i. $f(0.2) = \dots$ iii. $f(2) = \dots$
- ii. $f(35) = \dots$ iv. $f(4.5) = \dots$

c. Fungsi Saturating Linear

$$y = f(x) = \begin{cases} 1 & ; \quad x \geq 1 \\ x & ; \quad -1 \leq x \leq 1 \\ -1 & ; \quad x \leq -1 \end{cases}$$

Buatlah fungsi *.m file dari fungsi diatas

Catatlah hasil nya dengan menggunakan fungsi **satlin.m** untuk:

- i. $f(0.2) = \dots$ iii. $f(-0.3) = \dots$
- ii. $f(3.5) = \dots$ iv. $f(-5.5) = \dots$

d. Fungsi Sigmoid bipolar

$$y = f(x) = \frac{1 - e^{-x}}{1 + e^{-x}}$$

Buatlah fungsi *.m file dari fungsi diatas

Catatlah hasil nya dengan menggunakan fungsi **sigbip.m** untuk:

- | | |
|----------------------|------------------------|
| i. $f(0.2) = \dots$ | iii. $f(-0.3) = \dots$ |
| ii. $f(0.3) = \dots$ | iv. $f(-0.5) = \dots$ |

2. Buatlah fungsi **lhebb.m**

```
function [w,b]=lhebb(pm,pt)

%algoritma Hebb

%Input : pm = pola-pola masukan
%       pt = pola-pola target

%Output : bobot

% inisialisasi semua bobot nol

w= rand(1,length(pm(1,:)));

w= w-w

b=0;

for k=1:length(pm(:,1))

    disp('data ke'),k

    s= pm(k,:);

    x=s;

    t=pt(k,:);

    y=t';

    % Perbarui bobot

    w = w + (y * x)
```

```
        b = b + y  
    end
```

3. Buatlah fungsi **simhebb.m**

```
function y=simhebb(px,w)  
yin = px*w';  
if yin > 0  
    y=1;  
else  
    y=-1;  
end
```

7.5. TUGAS

Jalankan program **prak7.m** berikut:

```

%Nama file: prakt7.m

%=====

%Mode training

clc

clear

pm=[-1 -1;-1 1; 1 -1;1 1];
pt=[-1; 1; 1; 1];
w=lhebb(pm,pt)

save bobot.mat w

%Mode testing

%-----

Px=input('Masukkan pola data yang diujikan: ')

load bobot.mat

w=bobot

y=simhebb(Px,w)

```

Catat hasilnya

<p style="text-align: center;">Nilai</p>	<p style="text-align: center;">Yogyakarta, Paraf asisten <.....></p>

PRAKTIKUM 8: ALGORITMA PERCEPTRON UNTUK PENGENALAN POLA

Pertemuan ke : 8

Total Alokasi Waktu : 90 menit

- Pre-Test : 10 menit
- Praktikum : 65 menit
- Post-Test : 15 menit

Total Skor Penilaian : 100%

- Pre-Test : 25 %
 - Praktikum : 40 %
 - Post-Test : 35%
-

8.1. TUJUAN DAN INDIKATOR CAPAIAN

Setelah mengikuti praktikum ini mahasiswa diharapkan:

1. Mahasiswa mampu memahami algoritma perceptron

Indikator ketercapaian diukur dengan: (sesuaikan dengan RPS)

1. Memahami konsep perceptron untuk pengenalan pola

8.2. TEORI PENDUKUNG

Perceptron biasanya digunakan untuk mengklasifikasikan pola tertentu dan juga dikenal sebagai pemisah secara linear. Perceptron mempunyai 3 lapis neuron yaitu: unit sensori, unit asosiasi dan unit tanggapan.

Output perceptron: $y = f(y_{in})$

dengan fungsi aktivasi:

$$f(y_{in}) = \begin{cases} 1 & \text{bila } y_{in} > \theta \\ 0 & \text{bila } -\theta \leq y_{in} \leq \theta \\ -1 & \text{bila } y_{in} < -\theta \end{cases}$$

Implementasi dalam MATLAB


```

function y=bipolar2(x,th)

    if x > th

        y=1;

    elseif x < -1*th

        y=-1;

    else

        y=0;

    end

end

end

```

Perubahan bobot:

$$w_i(\text{baru}) = w_i(\text{lama}) + \alpha \cdot t \cdot x_i$$

dengan nilai target $t = +1$ atau -1 dan $\alpha =$ laju pelatihan

Algoritma Perceptron (MISO/Multi Input Single Output):

L0. Inisialisasi bobot dan bias (untuk mudahnya tetapkan semua bobot dan bias sama dengan nol)

Tetapkan laju pelatihan α ($0 < \alpha \leq 1$) dan (untuk mudahnya $\alpha = 1$)

L1. Selama syarat berhenti bernilai *false*, kerjakan langkah 2-6,

L2. Untuk setiap pasangan pembelajaran $s:t$, kerjakan langkah 3-5,

L3. Tetapkan aktivasi unit input

$$x_i = s_i; \quad \text{dengan } i = 1, 2, \dots, n$$

L4. Hitung tanggapan unit output

$$y_{in} = b + \sum_{i=1}^n x_i w_i$$

$$y = \begin{cases} 1; & \text{jika } y_{in} > \theta \\ 0; & \text{jika } -\theta \leq y_{in} \leq \theta \\ -1; & \text{jika } y_{in} < -\theta \end{cases}$$

L5. Perbarui bobot dan bias jika terjadi *error*:

Bila $y \neq t$ maka:

$$w_i(\text{baru}) = w_i(\text{lama}) + \alpha \cdot t \cdot x_i$$

$$b(\text{baru}) = b(\text{lama}) + \alpha \cdot t$$

Bila $y = t$, maka:

$$w_i(\text{baru}) = w_i(\text{lama})$$

$$b(\text{baru}) = b(\text{lama})$$

L6. Uji syarat berhenti; jika tidak terjadi perubahan bobot pada langkah 2, maka kondisi berhenti *true*, namun bila masih terjadi perubahan maka kondisi berhenti *false*.

```
function [w,b,err]=lperceptron1(pm,pt,lp,maxiter)

%Input : pm = pola-pola masukan
%        pt = pola-pola target
%        lp = laju pembelajaran
%        maxiter = jumlah maksimum iterasi
%        eps = toleransi error

%Output : w = bobot
%         b = bias
%         err = error

%=====

% inisialisasi semua bobot dan bias nol
w= rand(1,length(pm(1,:)));
w=w-w;
b=0;
n=0;
MSE=1;
err=[];
while (n < maxiter)
    disp('iterasi ke:'),n+1
    for k=1:length(pm(:,1))
        disp('data ke:'),k
        s= pm(k,:);
        x=s;
```

```

        yin=b+sum(x*w');
        y=bipolar2(yin,0.2);
        t=pt(k,:);
        if y ~= t
            % Perbarui bobot
            w= w + (lp*t* x)
            b= b+(lp*t)
        else
            % Perbarui bobot
            w=w
            b=b
        end

        e(k)=(y-t)
    end

    n=n+1;
    err(n)=sqrt(sum(e.^2))/(length(pm(:,1))*n);
    MSE=err(n)

end

```

Algoritma Perceptron MIMO (Multi Input Multi Output):

L0. Inisialisasi bobot dan bias (untuk mudahnya tetapkan semua bobot dan bias sama dengan nol)

Tetapkan laju pelatihan α ($0 < \alpha \leq 1$) dan (untuk mudahnya $\alpha = 1$)

L1. Selama syarat berhenti bernilai *false*, kerjakan langkah 2-6, {jumlah iterasi / toleransi error}

L2. Untuk setiap pasangan pembelajaran s:t, kerjakan langkah 3-5,

L3. Tetapkan aktivasi unit input

$$x_i = s_i; \quad \text{dengan } i = 1, 2, \dots, n$$

L4. Hitung tanggapan unit output

$$y_{in_j} = b_j + \sum_{i=1}^n x_i w_{ij}$$

$$y_j = \begin{cases} 1; & \text{jika } y_in_j > \theta \\ 0; & \text{jika } -\theta \leq y_in_j \leq \theta \\ -1; & \text{jika } y_in_j < -\theta \end{cases}$$

L5. Perbarui bobot dan bias jika terjadi *error*:

Bila $y \neq t$ maka:

$$w_{ij}(\text{baru}) = w_{ij}(\text{lama}) + \alpha.t_j.x_i$$

$$b_j(\text{baru}) = b_j(\text{lama}) + \alpha.t_j$$

Bila $y = t$, maka:

$$w_{ij}(\text{baru}) = w_{ij}(\text{lama})$$

$$b(\text{baru}) = b(\text{lama})$$

L6. Uji syarat berhenti; jika tidak terjadi perubahan bobot pada langkah 2, maka kondisi berhenti *true*, namun bila masih terjadi perubahan maka kondisi berhenti *false*.

```
function [w,b,err]=lperceptron2(pm,pt,lp,maxiter)

%Input : pm = pola-pola masukan

%      pt = pola-pola target

%      lp = laju pembelajaran

%      maxiter = jumlah maksimum iterasi

%      eps = toleransi error

%Output : w = bobot

%      b = bias

%      err = error

%=====

% inisialisasi semua bobot dan bias nol

n=length(pm(1,:));

m=length(pt(1,:));

w= zeros(n,m);

b=zeros(1,m);

err=[];

while (n < maxiter);
```

```

for iter=1:maxiter

    disp('iterasi ke:'),iter

    for k=1:length(pm(:,1))

        disp('data ke:'),k

        for i=1:n;

            s(i)= pm(k,i);

            x(i)=s(i);

        end

        for j=1:m;

            yin(j) =b(j)+sum(x(i)*w(i,j)');

            y(j)=bipolar2(yin(j),0.5);

        end

        if y(j) ~= t(j)

            % Perbarui bobot

            w(i,j)= w(i,j) + (lp*t(j)* x(i,j))

            b(j)= b(j)+(lp*t(j))

        else

            % Perbarui bobot

            w(i,j)=w(I,j)

            b(j)=b(j)

        end

    end

end

end

```

8.3. ALAT DAN BAHAN

Alat dan bahan yang digunakan dalam praktikum ini yaitu:

1. Komputer.
2. Software Matlab

8.4. LANGKAH PRAKTIKUM

1. Lakukanlah pelatihan dan pengujian fungsi AND dengan laju pembelajaran $\alpha = 1$ dan iterasi 3. Buat dan jalankan **Prak81a.m**

```
%Prak81a.m

%Data pelatihan

%-----

clear

pm=[1 1;1 0;0 1;0 0];
pt=[1;0;0;0];

%Mode training

% inisialisasi semua bobot nol

[w,b,err]=lperceptron2(pm,pt,1,3)

save bobotp.mat w b
```

Buat dan jalankan **Prak81b.m** dengan memberikan masukan masing-masing data yang digunakan pelatihan.

```
%Prak81b.m

%Mode testing

%-----

clear

load bobotp.mat

Px=input('Masukkan pola data yang diujikan: ')

yin=b+Px*w'

Th=1; % pada percobaan diganti 2, 3,...

y=bipolar2(yin, Th)
```

Isilah tabel hasil percobaan berikut:

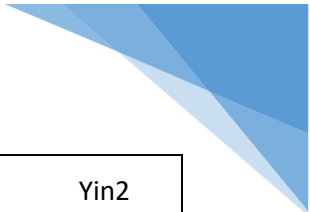
x1	x2	yin	Th =1	Th = 2	Th = 3
			y	y	y
1	1				
1	0				
0	1				
0	0				

Dari hasil percobaan diperoleh bahwa data yang telah dilatihkan akan dapat dikenali dengan baik bila dipilih nilai threshold

2. Dengan cara seperti pada percobaan 1 diatas, lakukan pelatihan dan pengujian untuk pasangan data pola masukan dan pola target dibawah ini.

pm (pola masukan)				pt (pola target)	
x1	x2	x3	x4	t1	t2
0	0	0	0	1	0
0	0	0	1	1	0
0	0	1	0	1	0
0	0	1	1	1	0
1	1	0	0	0	1
1	1	0	1	0	1
1	1	1	0	0	1
1	1	1	1	0	1

- a. Buatlah program dengan nama file **Prak82a.m** dengan cara memodifikasi program **Prak81a.m** untuk proses pelatihan dan gunakan fungsi **lperceptron2.m** dan berikan iterasi = 10.
- b. Buatlah program dengan nama file **Prak82b.m** dengan cara memodifikasi program **Prak81b.m** untuk proses pengujian (testing)
- c. Isilah tabel hasil percobaan berikut:



x1	x2	x3	x4	Yin1	Yin2
0	1	0	0		
0	0	0	1		
0	1	1	0		
0	0	1	1		
1	1	0	0		
1	0	0	1		
1	1	1	0		
1	0	1	1		

Tentukan nilai threshold yang dipilih agar data pelatihan dapat dikenali dengan baik.

8.5. TUGAS

<p>Nilai</p>	<p>Yogyakarta,</p> <p>.....</p> <p>Paraf asisten</p> <p><.....></p>

PRAKTIKUM 9: ALGORITMA BACKPROPAGATION UNTUK PENGENALAN POLA

Pertemuan ke : 9

Total Alokasi Waktu : 90 menit

- Pre-Test : 10 menit
- Praktikum : 65 menit
- Post-Test : 15 menit

Total Skor Penilaian : 100%

- Pre-Test : 25 %
- Praktikum : 40 %
- Post-Test : 35%

9.1. TUJUAN DAN INDIKATOR CAPAIAN

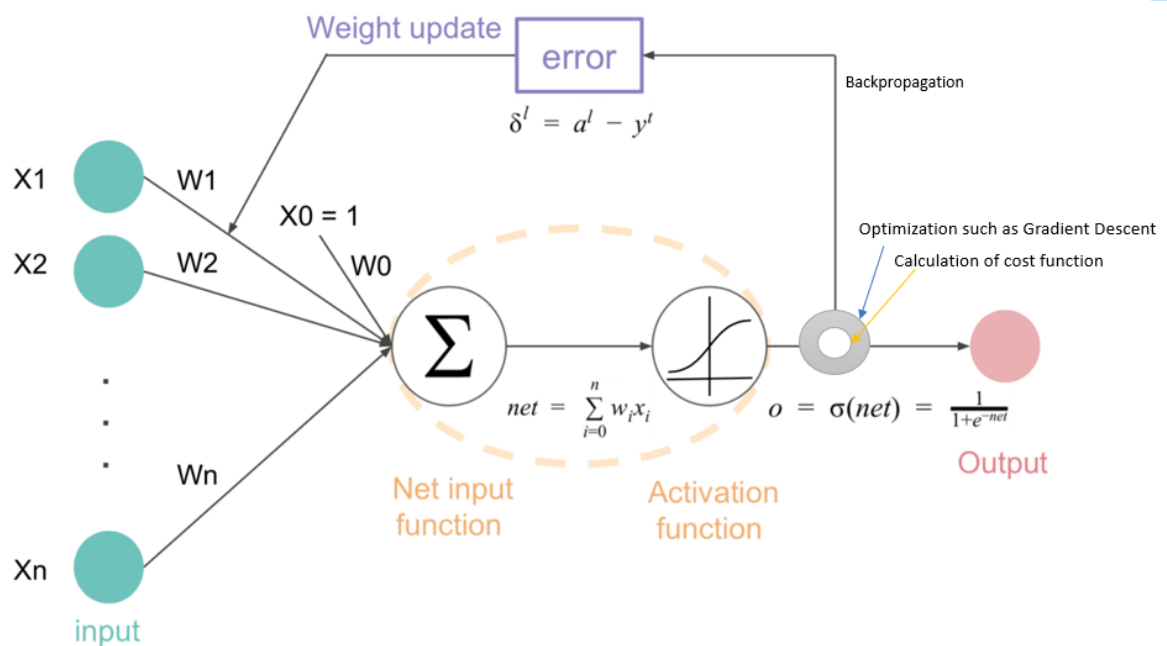
Setelah mengikuti praktikum ini mahasiswa diharapkan: Mahasiswa memahami konsep algoritma backpropagation

Indikator ketercapaian diukur dengan: Memahami konsep algoritma backpropagation untuk pengenalan pola

9.2. TEORI PENDUKUNG

Backpropagation (BP) merupakan algoritma pembelajaran yang terdiri dari tiga bagian yaitu: umpan maju pola pelatihan, perhitungan perambatan balik dan pengaturan bobot. Algoritma ini menggunakan *error* output untuk mengubah nilai bobot-bobotnya dalam arah balik (*backward*). Nilai *error* diperoleh dari proses sebelumnya yaitu pada tahap perambatan maju (*forward*). Inti dari algoritma belajar propagasi balik ini terletak pada kemampuannya mengubah nilai-nilai bobotnya untuk merespon adanya *error*.

Untuk dapat menghitung error, sampel data yang digunakan sebagai pembelajaran harus mengandung serangkaian pola-pola input beserta pasangan pola-pola output yang menjadi targetnya. Dengan pengertian lain bahwa perlu adanya pola-pola referensi bagi JST. Hal ini penting sehingga setiap JST mengeluarkan output, ia akan membandingkan dengan hasil yang diharapkan tadi. Propagasi balik *error* inilah yang memberi nama JST sebagai JST BP.



Gambar 9. 1 Arsitektur Backpropagation

Algoritma:

- L0. Inisialisasi bobot (tetapkan dengan nilai acak kecil)
- L1. Selama syarat kondisi *false* kerjakan langkah 2-9
- L2. Untuk setiap pasangan yang akan dilakukan pembelajaran, kerjakan langkah 3-8

Umpan maju

- L3. Setiap unit input ($x_i, i=1,2,\dots,n$) menerima sinyal input x_i dan meneruskan ke semua unit dalam lapis tersembunyi
- L4. Setiap unit tersembunyi ($z_j, j=1,2,\dots,p$) menjumlahkan sampel input terbobotnya,

$$z_in_j = v_{0j} + \sum_{i=1}^n x_i v_{ij}$$

Hitung sinyal output dengan fungsi aktivasinya

$$z_j = f(z_in_j)$$

Kirimkan sinyal ini ke semua unit pada lapisan output.

- L5. Setiap unit output ($y_k, k=1,2,\dots,m$) menjumlahkan sinyal input terbobotnya

$$y_in_k = w_{0k} + \sum_{j=1}^p z_j w_{jk}$$

Hitung sinyal output dengan fungsi aktivasinya

$$y_k = f(y_in_k)$$

Perambatan balik

L6. Setiap unit output ($y_k, k=1, \dots, m$) menerima pola target yang sesuai dengan pola input pembelajaran, hitung suku informasi *error* nya

$$\delta_k = (t_k - y_k) f'(y_in_k)$$

Hitung suku koreksi bobot (digunakan untuk perbaruan w_{jk}),

$$\Delta w_{jk} = \alpha \delta_k z_j$$

Hitung suku koreksi bias (digunakan untuk perbaruan w_{ok}),

$$\Delta w_{ok} = \alpha \delta_k$$

Kirimkan δ_k ke unit-unit dilapis bawahnya

L7. Setiap unit tersembunyi ($z_j, j=1, \dots, p$) menjumlahkan delta inputnya (dari unit-unit yang berada pada lapis di atasnya)

$$\delta_in_j = \sum_{k=1}^m \delta_k w_{jk}$$

Hitung suku informasi *error*,

$$\delta_j = \delta_in_j f'(z_in_j)$$

Hitung suku koreksi bobot (untuk perbaruan v_{ij})

$$\Delta v_{ij} = \alpha \delta_j x_i$$

Hitung suku koreksi bias (untuk perbaruan v_{oj}),

$$\Delta v_{oj} = \alpha \delta_j$$

Perbaruan bobot dan bias:

L8. Setiap unit output ($y_k, k=1, \dots, m$) perbarui bobot-bobot dan bias nya:

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk}$$

Setiap unit tersembunyi ($z_j, j=1, \dots, p$) perbarui bobot-bobot dan bias nya ($i=0, \dots, n$):

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij}$$

L9. Uji syarat berhenti

Catatan:

Fungsi aktivasi sigmoid

$$f(x) = \frac{1}{1 + e^{-x}}$$

$$f'(x) = f(x)[1 - f(x)]$$

9.3. ALAT DAN BAHAN

Alat dan bahan yang digunakan dalam praktikum ini yaitu:

1. Komputer.
2. Software Matlab

9.4. LANGKAH PRAKTIKUM

1. Lakukanlah pelatihan dan pengujian fungsi XOR dengan laju pembelajaran $\alpha = 0.05$ dan iterasi 500. Buat dan jalankan **Prak91a.m** (menggunakan fungsi toolbox NN)

```
%Prak91a.m
%Data pelatihan
%-----
pm1=[0 0]';
pm2=[0 1]';
pm3=[1 0]';
pm4=[1 1]';
pm=[pm1 pm2 pm3 pm4]
pt=[0 1 1 0]
%mode training
net=newff(minmax(pm),[2,1],{'tansig','tansig'},'traingd');
net.trainParam.show = 50;
net.trainParam.lr = 0.5;
net.trainParam.epochs = 500;
net.trainParam.goal = 1e-5;
[net,tr]=train(net,pm,pt);
%mode testing
p=input('Masukkan pola : ')
[yin]= sim(net,p)
```

2. Contoh sistem pengenalan wajah

```
%Prak92.m
%-----
clc
clear
% Membaca data wajah untuk pelatihan terdiri dari 5 wajah orang yang berbeda
% dengan masing-masing menggunakan 4 sampel data
for i = 1:4
```

```

data1{i} = imread(['C:\Program Files\MATLAB704\work\face\f1\' num2str(i) '.tif']);
data2{i} = imread(['C:\Program Files\MATLAB704\work\face\f2\' num2str(i) '.tif']);
data3{i} = imread(['C:\Program Files\MATLAB704\work\face\f3\' num2str(i) '.tif']);
data4{i} = imread(['C:\Program Files\MATLAB704\work\face\f4\' num2str(i) '.tif']);
data5{i} = imread(['C:\Program Files\MATLAB704\work\face\f5\' num2str(i) '.tif']);
end;

```

% Pemrosesan awal dan ekstraksi ciri

```
for i = 1:4
```

```

    data1{i} = imresize(data1{i},[30 20]);
    x1{i}=reshape(data1{i},[600,1]);
    x1{i}=x1{i}-mean(x1{i});
    x1{i}=x1{i}/max(x1{i});

```

```

    data2{i} = imresize(data2{i},[30 20]);
    x2{i}=reshape(data2{i},[600,1]);
    x2{i}=x2{i}-mean(x2{i});
    x2{i}=x2{i}/max(x2{i});

```

```

    data3{i} = imresize(data3{i},[30 20]);
    x3{i}=reshape(data3{i},[600,1]);
    x3{i}=x3{i}-mean(x3{i});
    x3{i}=x3{i}/max(x3{i});

```

```

    data4{i} = imresize(data4{i},[30 20]);
    x4{i}=reshape(data4{i},[600,1]);
    x4{i}=x4{i}-mean(x4{i});
    x4{i}=x4{i}/max(x4{i});

```

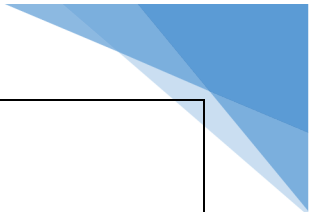
```

    data5{i} = imresize(data5{i},[30 20]);
    x5{i}=reshape(data5{i},[600,1]);
    x5{i}=x5{i}-mean(x5{i});
    x5{i}=x5{i}/max(x5{i});

```

```
end;
```

```
%F1
```



```
pm11=double(x1{1});  
pm12=double(x1{2});  
pm13=double(x1{3});  
pm14=double(x1{4});
```

```
%F2
```

```
pm21=double(x2{1});  
pm22=double(x2{2});  
pm23=double(x2{3});  
pm24=double(x2{4});
```

```
%F3
```

```
pm31=double(x3{1});  
pm32=double(x3{2});  
pm33=double(x3{3});  
pm34=double(x3{4});
```

```
%F4
```

```
pm41=double(x4{1});  
pm42=double(x4{2});  
pm43=double(x4{3});  
pm44=double(x4{4});
```

```
%F5
```

```
pm51=double(x5{1});  
pm52=double(x5{2});  
pm53=double(x5{3});  
pm54=double(x5{4});
```

```
pm=[pm11 pm12 pm13 pm14 pm21 pm22 pm23 pm24 pm31 pm32 pm33 pm34 pm41  
pm42 pm43 pm44 pm51 pm52 pm53 pm54];
```

```
pt1=[0 0 0 0 1]';
```

```
pt2=[0 0 0 1 0]';
```

```
pt3=[0 0 1 0 0]';
```

```
pt4=[0 1 0 0 0]';
```

```

pt5=[1 0 0 0 0]';

pt=[pt1 pt1 pt1 pt1 pt2 pt2 pt2 pt2 pt3 pt3 pt3 pt3 pt4 pt4 pt4 pt4 pt5 pt5 pt5 pt5];

% jumlah neuron pada lapis tersembunyi 300
net=newff(minmax(pm),[300,5],{'tansig','tansig'},'traingd');
net.trainParam.show = 50;
net.trainParam.lr = 0.5;
net.trainParam.epochs = 500;
net.trainParam.goal = 1e-6;
[net,tr]=train(net,pm,pt);
%testing
Im = imread('C:\Program Files\MATLAB704\work\face\f1\5.tif');
I = imresize(Im,[30 20]);
x=reshape(I,[600,1]);
x=double(x-mean(x));
p=x/max(x);

[yin]= sim(net,p)

```


9.5. TUGAS

1. Lakukan proses pelatihan dan pengujian dengan variasi parameter:

- Jumlah neuron pada lapis tersembunyi 300
- Laju pembelajaran 0,5
Epoch: 500 dan 1000

Hasil Percobaan 1:

Masukan	Jumlah neuron lapis tersembunyi : 300 Epoch:500, laju pembelajaran: 0,5				
	F1	F2	F3	F4	F5
F1					
F2					
F3					
F4					



F5					
Akurasi (%)					

Hasil Percobaan 2:

Masukan	Jumlah neuron lapis tersembunyi : 300 Epoch:1000, laju pembelajaran: 0,5				
	F1	F2	F3	F4	F5
F1					
F2					
F3					
F4					
F5					
Akurasi (%)					

<div> <div>Nilai</div> </div>	<div> <div>Yogyakarta,</div> <div>.....</div> <div>Paraf asisten</div> <div><.....></div> </div>

PRAKTIKUM 10: ALGORITMA LVQ UNTUK PENGENALAN POLA

Pertemuan ke : 10

Total Alokasi Waktu : 90 menit

- Pre-Test : 10 menit
- Praktikum : 65 menit
- Post-Test : 15 menit

Total Skor Penilaian : 100%

- Pre-Test : 25 %
- Praktikum : 40 %
- Post-Test : 35%

10.1 TUJUAN DAN INDIKATOR CAPAIAN

Setelah mengikuti praktikum ini mahasiswa diharapkan: Mahasiswa mampu memahami algoritma LVQ

Indikator ketercapaian diukur dengan: Mahasiswa mampu memahami algoritma LVQ untuk pengenalan pola

10.2 TEORI PENDUKUNG

Metode pembelajaran ini mengatur batas-batas antara kategori-kategori yang berbeda untuk menjaga kesalahan klasifikasi seminimal mungkin. Pembelajaran dilakukan pada lapis kompetitif yang terawasi. Suatu lapis kompetitif akan secara otomatis belajar untuk mengklasifikasi vektor-vektor input. Klas-klas yang didapatkan sebagai hasil dari lapisan kompetitif ini hanya tergantung pada jarak antara vektor-vektor input.

Algoritma:

- x** : vektor pelatihan ($x_1, \dots, x_i, \dots, x_n$)
- T** : klas atau kategori vektor pembelajaran
- w_j** : vektor bobot unit output; ($w_{1j}, \dots, w_{ij}, \dots, w_{nj}$)
- C_j** : kategori atau klas yang berikan oleh unit output j
- $\|x - w_j\|$: jarak euclidean antara vektor input dan vektor

L0. Inisialisasi vektor referensi

 Inisialisasi laju pembelajaran $\alpha(0)$

L1. Selama syarat berhenti *false* kerjakan langkah 2-6

L2. Untuk setiap vektor input pembelajaran x, kerjakan langkah 3-4

L3. Cari J sehingga $\|x - w_j\|$ minimum

L4. Perbarui w_j :

Bila $\mathbf{T} = \mathbf{C}_j$, maka

$$w_j(\text{baru}) = w_j(\text{lama}) + \alpha \cdot [x - w_j(\text{lama})]$$

Bila $\mathbf{T} \neq \mathbf{C}_j$, maka

$$w_j(\text{baru}) = w_j(\text{lama}) - \alpha \cdot [x - w_j(\text{lama})]$$

L5. Kurangi laju pembelajaran

L6. Uji syarat berhenti;

Syaratnya: spesifikasi cacah iterasi atau laju pembelajaran mencapai nilai yang cukup kecil.

10.3 ALAT DAN BAHAN

Alat dan bahan yang digunakan dalam praktikum ini yaitu:

1. Komputer.
2. Software Matlab

10.4 LANGKAH PRAKTIKUM

1. Jalankan program berikut:

```
%Prak10.m

p1=[1;1;0;0];
p2=[0;0;0;1];
p3=[0;0;1;1];
p4=[1;0;0;0];
p5=[0;1;1;0];

P = [p1 p2 p3 p4 p5];
C = [1 2 1 2 2];
T = ind2vec(C);

%mode training
net = newlvq(minmax(P),4,[0.6 0.4],0.01);
net=train(net,P,T);
```

```
%mode testing

p = input('Masukkan pola baru :')

a = vec2ind(sim(net,p))
```

2. Implementasikan sistem pengenalan wajah pada Praktikum 9 dengan menggunakan algoritma LVQ.

Hasil percobaan 1:

Masukan	Jumlah neuron lapis tersembunyi : 300				
	F1	F2	F3	F4	F5
F1					
F2					
F3					
F4					
F5					
Akurasi (%)					

Hasil percobaan 2:

Masukan	Jumlah neuron lapis tersembunyi : 600				
	F1	F2	F3	F4	F5
F1					
F2					
F3					
F4					
F5					



Akurasi (%)	
----------------	--

10.5 TUGAS

Nilai	Yogyakarta, Paraf asisten <.....>



DAFTAR PUSTAKA

1. Tou, Julius T and Gonzalez, Rafael C., *Pattern Recogniton Principles*, Addison-Wesley Publishing Company, London
2. Schalkoff, R.J., 1992, *Pattern Recognition: Statistical, Structural and Neural Approaches*, John Wiley & Sons, Inc., Singapore
3. Duda R.O., Hart P.E., Stork D.G., 2000, *Pattern classification*, 2ed., Wiley, ISBN 0471056693
4. Fausett, L., 1994, *Fundamentals of Neural Networks: architecture, algoritma, and applications*, Prentice Hall, New Jersey.
5. Sivanandam,S.N., Sumathi, S., and Deepa, S.N., 2006, ***Introduction to Neural Networks using Matlab 6.0***, The McGraw-Hill Comp.

